

UNIVERSIDAD DE ALMERIA

ESCUELA SUPERIOR DE INGENIERÍA

“Aplicación híbrida para memorizar usando
repetición espaciada”

Mención:

Ingeniería del Software/Tecnologías de la
Información

Curso 2019/2020

Alumno/a:

Juan Miranda Berenguel

Director/es:

José Joaquín Cañadas Martínez



Aplicación híbrida para memorizar usando repetición espaciada

Juan Miranda Berenguel
Junio 2020

para minha mãe...

Agradecimientos

En primer lugar, quiero agradecer a mis padres y en especial a mi madre por transmitirme esa constancia y hacerme ver que todo se puede conseguir.

Asimismo, también quiero agradecer a la Universidad de Almería y a todos sus docentes por las enseñanzas aportadas durante estos años. Quiero destacar en especial a mi tutor Joaquín Cañadas por su apoyo en esta última etapa.

Por ultimo y no menos importante a mis amigos y pareja porque por vosotros mereció la pena.

obrigado pelo vosso apoio...

ÍNDICE DE CONTENIDO

Índice de Tablas	v
Índice de Ilustraciones	vii
Resumen.....	ix
Abstract.....	xi
Convenciones Utilizadas	xiii
1 MEMORIA DESCRIPTIVA DEL PROYECTO	1
1.1 Introducción	1
1.2 Motivación para el desarrollo.....	2
1.3 Objetivos.....	3
1.4 Planificación temporal	3
1.5 Tecnologías utilizadas.....	4
1.6 Estructura del documento	4
2 FASES DE LA REALIZACIÓN DEL TRABAJO.....	7
2.1 Introducción	7
2.2 Metodología de trabajo.....	8
2.2.1 Elección de la metodología de trabajo.....	8
2.2.2 Introducción al Proceso Unificado Rational - RUP.....	9
2.3 Planificación del proyecto.....	10
2.3.1 Planificación de las etapas.....	10
2.3.2 Etapa inicial.....	11
2.3.3 Etapa de Elaboración	11
2.3.4 Etapa de Construcción	12
2.3.5 Etapa de Transición	15
2.4 Seguimiento del proyecto.	16
3 ESTADO DEL ARTE	19
3.1 Introducción	19
3.2 La curva del olvido y el repaso espaciado	20
3.2.1 Técnicas de repaso espaciado	21
3.3 Tecnologías digitales de aprendizaje	22
3.3.1 Efecto de la tecnología de aprendizaje digital.....	22

3.3.2	Dispositivos electrónicos utilizados para estudiar.....	23
3.3.3	Tipos de tecnología digital que utilizan los estudiantes	24
3.3.4	Impacto del análisis del rendimiento en el aprendizaje.....	25
3.4	Aplicaciones referentes de repaso espaciado	27
3.4.1	Duolingo	27
3.4.2	Menrise.....	28
3.4.3	Anki	28
3.4.4	Lingvist.....	29
3.4.5	SuperMemo.....	29
3.5	Comparativas aplicaciones referentes.....	30
3.6	Crítica al estado del arte, conclusiones y propuesta.....	30
4	TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS	33
4.1	Introducción	33
4.2	Aplicaciones Híbridas.....	34
4.3	Aplicaciones Web Progresivas - PWA	34
4.4	Framework empleados en la construcción del front-end.....	37
4.4.1	Apache Cordova.....	37
4.4.2	Electron Framework	38
4.4.3	Quasar Framework.....	38
4.5	Lenguajes de programación empleados	39
4.5.1	Introducción	39
4.5.2	Vue	40
4.5.3	JavaScript	45
4.5.4	HTML.....	46
4.5.5	CSS.....	47
4.6	Firesabase como "back-end as a service"	48
4.6.1	Introducción	48
4.6.2	Firebase.....	49
4.7	Tecnologías utilizadas para la persistencia de datos	50
4.7.1	Antecedentes para la elección de la base de datos.....	50
4.7.2	Cloud Firestore, el motor y la base de datos	51
4.8	APIS utilizadas en el sistema.....	52
4.8.1	Pixabay API	52
4.8.2	YouTube Data API	52
4.9	Herramientas utilizadas para el desarrollo.....	53
4.9.1	Git	53

4.9.2	Visual Studio Code	54
4.9.3	Balsamiq Mockups.....	55
4.9.4	Visual Paradigm Online	55
5	ANÁLISIS DEL SISTEMA	57
5.1	Introducción	57
5.2	Especificación preliminar del sistema.....	58
5.3	Objetivos del sistema	59
5.4	Requisitos del sistema.....	60
5.4.1	Requisitos de información	60
5.4.2	Requisitos funcionales	65
5.4.3	Requisitos no funcionales	82
5.4.4	Matrices de trazabilidad	83
6	DISEÑO Y CONSTRUCCIÓN DEL SISTEMA	85
6.1	Introducción	85
6.2	Arquitectura del sistema.....	86
6.3	Componentes de un solo archivo	90
6.4	Patrón de gestión de estado.....	93
6.5	Patrón Enrutador	94
6.6	El modelo de clases	96
6.7	El Modelo de almacenamiento de datos	97
6.7.1	Introducción	97
6.7.2	Esquema de datos.....	97
6.7.3	Colecciones de datos	100
6.7.4	Seguridad de los datos	103
6.8	Convención de nombres y estructura de directorios	106
6.9	Tipos de usuarios en el sistema	107
6.10	Prototipo Inicial	108
6.10.1	Pantalla inicial.....	109
6.10.2	Pantalla de progreso	109
6.10.3	Pantalla de cursos	110
6.10.4	Pantalla de tarjeta de aprendizaje	111
6.11	Paleta de colores y tipología de letra empleada.....	112
6.12	Implementación del Algoritmo de repetición espaciada	113
6.12.1	Introducción a la planificación de la repetición	113
6.12.2	El algoritmo SM-2 genérico.....	113
6.12.3	Modificaciones efectuadas del algoritmo SM-2.....	116

6.12.4	Atributos para la planificación de la repetición.....	116
6.12.5	Flujograma de planificación de repeticiones	117
6.13	Las tarjetas de aprendizaje	119
6.13.1	Tarjetas de tipo test	120
6.13.2	Tarjetas de tipo subtítulos de video	122
6.13.3	Generar nuevas modalidades de tarjetas de aprendizaje ...	127
6.14	Despliegue.....	130
6.14.1	El backend	130
6.14.2	El frontend en diferentes plataformas y dispositivos.....	130
7	PRUEBAS Y CONTROL DE CALIDAD.....	137
7.1	Introducción	137
7.2	Plan de Pruebas.....	138
7.2.1	Definición de los casos de prueba.....	138
7.2.2	Trazabilidad de los casos de prueba.....	142
7.3	Control de calidad.....	143
7.3.1	Introducción	143
7.3.2	Métricas de Accesibilidad.....	144
7.3.3	Métricas de tiempo de actuación	145
7.3.4	Métricas sobre el uso de buenas prácticas	146
7.3.5	Métricas aplicación Web Progresiva	147
7.3.6	Métricas de posicionamiento SEO	148
8	CONCLUSIONES Y TRABAJO FUTURO.....	149
8.1	Introducción	149
8.2	Conclusiones.....	150
8.3	Líneas de Mejora	151
	BIBLIOGRAFÍA.....	153
ANEXOS		
	APÉNDICE A - MANUAL DE USUARIO	159
A.1	Introducción	159
A.2	Autenticación y registro en el sistema	160
A.3	Pantalla principal.....	162
A.4	Pantalla de progreso	168
A.5	Pantalla de cursos	169
A.6	Pantalla perfil	184

ÍNDICE DE TABLAS

Tabla 2-1: Planificación de fases del proyecto.....	10
Tabla 2-2: Iteraciones y tareas de la fase inicial	11
Tabla 2-3: Iteraciones y tareas de la fase de elaboración	12
Tabla 2-4: Iteraciones C1-C3 fase de construcción.....	13
Tabla 2-5 Iteraciones C4-C6 fase de construcción	13
Tabla 2-6: Iteraciones C7-C8 fase de construcción.....	14
Tabla 2-7: Iteraciones y tareas de la fase de transición	15
Tabla 3-1: Características Duolingo.....	27
Tabla 3-2: Características Memrise	28
Tabla 3-3: Características Anki	28
Tabla 3-4: Características Lingvist	29
Tabla 3-5: Características SuperMemo	29
Tabla 3-6: Comparativa aplicaciones referentes.....	30
Tabla 4-1: Características Cordova	37
Tabla 4-2: Características Electron.....	38
Tabla 4-3: Características Quasar	38
Tabla 4-4: Características de JavaScript	45
Tabla 4-5: Características de HTML.....	46
Tabla 4-6: Características de HTML.....	47
Tabla 4-7: Características Firebase	49
Tabla 4-8: Diferencias de esquemas de una misma colección	51
Tabla 4-9: Características Cloud Firestore [31].....	52
Tabla 4-10: Características Git	53
Tabla 4-11: Características GitLab	54
Tabla 4-12: Características VS Code	54
Tabla 4-13: Características Balsamiq.....	55
Tabla 5-1: Matriz de trazabilidad de requisitos de información frente a objetivos generales.....	83
Tabla 5-2: Matriz de trazabilidad de casos de uso frente a objetivos Generales	84
Tabla 5-3: Matriz de trazabilidad de casos de uso frente a requisitos de información.....	84
Tabla 6-1: Ejemplo de enrutamiento de la App	95
Tabla 6-2: Estructuración Jerárquica Colección Users	101
Tabla 6-3: Estructuración Jerárquica Colección Courses.....	101
Tabla 6-4: Estructuración Jerárquica Colección Units	102
Tabla 6-5: Estructuración Jerárquica Colección Cards	103
Tabla 6-6: Reglas de seguridad de datos	104
Tabla 6-7: Regla de seguridad de datos RSD-001	105
Tabla 6-8: Regla de seguridad de datos RSD-008	105

Tabla 6-9: Estructuración de los archivos del proyecto	106
Tabla 6-10: Convención de nombres	107
Tabla 6-11: Tipos de usuarios y permisos	108
Tabla 6-12: Secuencia de acciones algoritmo SM-2	115
<i>Tabla 6-13: XML subtítulos API Youtube</i>	<i>125</i>
Tabla 6-14: Fichero CCreateCardTest.vue	128
Tabla 6-15: Método Completar tarjetas	129
Tabla 7-1: Matriz de trazabilidad de Requisitos – Caso de prueba	143
Tabla 7-2: Datos auditoria accesibilidad.....	145
Tabla 7-3: Datos auditoria tiempos de actuación	146
Tabla 7-4: Datos auditoria buenas prácticas.....	147
Tabla 7-5: Datos auditoria PWA.....	148

ÍNDICE DE ILUSTRACIONES

Ilustración 1-1: Cronograma etapas del TFG en semanas.....	3
Ilustración 2-1 Grafico Hump metodología RUP.....	9
Ilustración 2-2 Calendario del recurso Juan asignado al proyecto	10
Ilustración 2-3: Grafica de commits en repositorio	16
Ilustración 2-4: Commits por día de la semana.....	17
Ilustración 2-5: Relación de Commits por franja horaria.	17
Ilustración 3-1: Curva del olvido Ebbinghaus.....	20
Ilustración 3-2: Sistema de Leitner	21
Ilustración 3-3: Efecto de las DLT en las calificaciones de los estudiantes [20]	22
Ilustración 3-4: Utilidad de la tecnología de aprendizaje adaptativa para la retención de nuevos conceptos [20].....	23
Ilustración 3-5: Tipos de Dispositivos Electrónicos que Usan los Estudiantes para Estudiar [20]	24
Ilustración 3-6: Tipos de tecnología digital de aprendizaje que utilizan los estudiantes [20].....	25
Ilustración 3-7: Influencia de la información de rendimiento [20]	26
Ilustración 4-1: Arquitectura app hibrida vs nativa.....	34
Ilustración 4-2 La web y el Service Worker [22]	36
Ilustración 4-3 Páginas que se comunican con un Service Worker mientras el usuario está offline [22]	36
Ilustración 4-4 Service Worker que se comunica con un servidor después de que un usuario ha salido de la página [22].....	36
Ilustración 4-5: Lenguajes de programación utilizados en el proyecto.....	39
Ilustración 4-6: Interés a lo largo del tiempo Vue.js.....	41
Ilustración 4-7: Descargas Vue.js vs AngularJS	42
Ilustración 4-8: Resultado comparativa rendimiento Vue	43
Ilustración 4-9: Características Visual Paradigm Online.....	55
Ilustración 6-1: Arquitectura global de la APP.....	86
Ilustración 6-2: Arquitectura de la APP Lembra.....	87
Ilustración 6-3: Representación de componentes páginas y layout	88
Ilustración 6-4: Esquema de una petición	89
Ilustración 6-5: Componentes en la interfaz.....	90
Ilustración 6-6: Componente CCHheader.....	91
Ilustración 6-7: Vista del componente CcHeader.....	92
Ilustración 6-8: Utilización del componente CcHeader	92
Ilustración 6-9: Flujo Estado Vuex	94
Ilustración 6-10: Diagrama de clases	96
Ilustración 6-11: Modelo de datos Cloud Firestore.....	97
Ilustración 6-12: Diagrama de estructuración de almacenamiento de datos	99
Ilustración 6-13: Prototipo Inicial - Pantalla Inicial	109

Ilustración 6-14: Prototipo Inicial Pantalla de progreso	109
Ilustración 6-15: Prototipo Inicial - Pantalla Cursos	110
Ilustración 6-16: Prototipo Inicial - Pantalla de tarjeta aprendizaje	111
Ilustración 6-17: Prolongación de los intervalos de repetición [42]	114
Ilustración 6-18: Flujograma programación de repeticiones	118
Ilustración 6-19: Contenido común de tarjetas	119
Ilustración 6-20: Tarjeta tipo test	120
Ilustración 6-21: Asistente tarjetas tipo test.	121
Ilustración 6-22: Tarjeta de tipo video.....	122
Ilustración 6-23: Asistente tarjeta video 1	123
Ilustración 6-24: Asistente tarjeta video 2	124
Ilustración 6-25: Asistente tarjeta video 3	126
Ilustración 6-26: Asistente tarjeta video 4	127
Ilustración 6-27: Compilación plataformas Windows.....	131
Ilustración 6-28: Lembra en Windows 10.....	131
Ilustración 6-29: Compilación Linux	131
Ilustración 6-30: Lembra en Ubuntu.....	132
Ilustración 6-31: Compilación Android	132
Ilustración 6-32: Lembra en dispositivo Android	133
Ilustración 6-33: compilación IOs	133
Ilustración 6-34: Lembra en iPhone 8.....	134
Ilustración 6-35: Despliegue en Firebase Hosting.....	135
Ilustración 6-36: PWA Lembra alojada en Firebase hosting	135
Ilustración 6-37: Lembra PWA.....	136
Ilustración 7-1: Resultados Accesibilidad Aplicación	144
Ilustración 7-2: Métricas tiempos de actuación	145
Ilustración 7-3 Métricas sobre el uso de buenas prácticas	146
Ilustración 7-4: Métricas aplicación web progresiva.....	147
Ilustración 7-5: Métricas parámetros para el posicionamiento SEO.....	148

Juan Miranda Berenguel

Autor del proyecto

José Joaquín Cañadas Martínez

Director del proyecto

RESUMEN

Existen diferentes técnicas de aprendizaje, una de estas técnicas de aprendizaje es el repaso espaciado en la que se repasa a intervalos crecientes. Para empezar el sujeto se familiariza con la información para fijarla en su memoria, posteriormente se realizan repasos periódicos y, conforme se va consolidando la información en la memoria, los repasos se distanciarán más y más en el tiempo. Un gran número de investigaciones de la psicología cognitiva y educativa han demostrado que realizar un estudio de forma espaciada en el tiempo es una forma de mejorar la efectividad y la eficiencia del aprendizaje.

En este Trabajo Fin de Grado se desarrolla una aplicación multiplataforma destinada al aprendizaje mediante técnicas de repaso espaciado que favorecen la retención de contenido en la memoria a largo plazo. El dominio en el que se ha puesto en práctica ha sido el aprendizaje de idiomas y las pruebas tipo test para las que han sido diseñados un conjunto de ejercicios los cuales pueden ser creados y editados por los usuarios desde la propia aplicación.

Durante este Trabajo Fin de Grado se abarcará desde el estado de arte, hasta el desarrollo de la aplicación justificando la elección de las tecnologías, la metodología y los métodos de trabajo empleados.

PALABRAS CLAVE

Repetición espaciada/distribuida, pruebas, aprendizaje, educación, frameworks, HTML5, CSS3, Firebase, Cloud Firestore, Cordova, RUP, responsive, GCP, Google Cloud Platform, PWA, Electron, Visual Paradigm, Balsamiq Mockups, Gitlab, Git, Visual Studio Code.

Juan Miranda Berenguel
Project Author

José Joaquín Cañadas Martínez
Project Director

ABSTRACT

There are different learning techniques, one of these learning techniques is spaced repetition in which the review process is scheduled in increasing. That is, to begin with, the subject becomes familiar with the information to fix it in his memory, then periodic reviews are carried out and, as the information is consolidated in the memory, the reviews will distance themselves more and more over time. A large number of research in cognitive and educational psychology have demonstrated that conducting a study spaced over time is a way to improve the effectiveness and efficiency of learning.

This End-Of-Degree dissertation presents the development of a multiplatform application for learning through spaced repetition techniques that favor long-term memory retention. The domain in which it has been put into practice has been language learning and standardized tests for which a set of exercises have been designed which can be created and edited by users from the application itself.

Along this dissertation, are covered from the state of the art, to the development of the application justifying the choice of technologies, methodology and working methods used.

KEYWORDS

Spaced / distributed repetition, testing, learning, education, frameworks, HTML5, CSS3, Firebase, Cloud Firestore, Cordova, RUP, responsive, NoSQL, GCP, Google Cloud Platform, PWA, Electron Visual Paradigm, Balsamiq Mockups, Gitlab, Git, Visual Studio Code,

CONVENCIONES UTILIZADAS

En este Trabajo Fin de Grado se utilizan ciertas convenciones habituales en este tipo de documentos. En este apartado se explican estas convenciones y el modo de interpretarlas.

- Las expresiones importantes son resaltadas con un formato negrita del texto.
- Las referencias cruzadas del documento se representan con un formato de letra cursiva y una tipografía *Calibri Light*
- Las palabras o expresiones en otros idiomas diferentes al castellano son representados en cursiva y con una tipología de letra *Calibri Light*
- Los nombres de personas, aplicaciones, metodologías y siglas son representados en cursiva con tipología *Calibri Light*
- Las anotaciones al texto son representadas mediante una nota en la parte inferior de la página. Estas anotaciones son identificadas por un superíndice numérico inmediatamente después de la línea de texto.
- El código fuente, independientemente del lenguaje de programación en el que se muestre se encuentra representado mediante una tipografía de letra `Courier new`



Capítulo

1

MEMORIA DESCRIPTIVA DEL PROYECTO

1.1 Introducción

La RAE¹ define aprender como “Adquirir el conocimiento de algo por medio del estudio o de la experiencia. Fijar algo en la memoria.” Es sabido que el conocimiento adquirido no permanece inalterable en nuestra mente, ya que tras un tiempo los recuerdos tienden a desaparecer. Desde hace años existen diferentes aplicaciones que aprovechan técnicas de aprendizaje como es el **repaso espaciado en el tiempo** para fomentar la retención de contenido en la memoria, estas aplicaciones son utilizadas por una gran comunidad, principalmente para aprender idiomas mediante el uso de **tarjetas de aprendizaje** o *flash cards*².

El resultado del presente trabajo fin de grado es el desarrollo de una aplicación multiplataforma, que haciendo uso de tarjetas de aprendizaje y técnicas de repaso espaciado posibilite el aprendizaje mediante la realización de ejercicios. Estos serán repetidos por el usuario en un intervalo de tiempo determinado, según las respuestas obtenidas y basándose en la idea de que hay un momento ideal para practicar lo aprendido.

En este capítulo describiremos cuales han sido las motivaciones para el desarrollo de este proyecto y cuáles son los objetivos generales que se pretenden cumplir, además realizaremos una introducción sobre la planificación y cuál es la estructuración de este documento trabajo fin de grado.

¹ RAE: Real Academia Española, es una institución academia dedicada a la regularización lingüística.

² **Flash Card**: son tarjetas que contienen palabras, imágenes, símbolos o números en uno o ambos lados y se usan para adquirir conocimientos. Normalmente en un lado de cada tarjeta se escribe una pregunta y en el otro la respuesta, forzando al usuario a recordar la respuesta a la pregunta.

1.2 Motivación para el desarrollo

Durante los últimos años, la necesidad de adquirir nuevos conocimientos me hizo usuario habitual de aplicaciones de memorización. Utilicé estas aplicaciones para aprender vocabulario nuevo en diferentes idiomas y para prepararme exámenes tipo test. Como un usuario más, desconocía que detrás de estas aplicaciones existen algoritmos y técnicas para mejorar la retención del contenido aprendido.

Una estancia en la localidad de Bragança, al norte de Portugal, entre los años 2016 – 2017, me hizo comenzar a estudiar portugués y a utilizar aplicaciones de memorización de forma continuada, con el objetivo de adquirir nuevo vocabulario. Sin embargo, tras unas semanas de uso y al combinar estas aplicaciones de memorización con otras técnicas de estudio, encontraba la necesidad de adaptar el vocabulario de la aplicación a mis necesidades, sin ser yo el que, por obligatoriedad del diseño de este tipo de aplicaciones, tuviese que utilizar un contenido ya definido y estático. Esta problemática me hizo realizar una pequeña aplicación web, únicamente con la funcionalidad de repasar vocabulario de forma aleatoria y sin ninguna técnica de memorización.

Meses después y tras comenzar a estudiar otro idioma, comencé a tener la misma problemática, el contenido no se adaptaba a mis necesidades. Es por esto por lo que, tras consultar comentarios de los usuarios de estas aplicaciones, son numerosas las opiniones que hacen referencia a esta problemática y que solicitan a los creadores de las principales aplicaciones que habiliten la posibilidad de crear contenido adaptado.

Después de una primera investigación, la cual dio lugar a la realización del anteproyecto de este trabajo fin de grado, concluí que las aplicaciones que fomentan el aprendizaje se encuentran en auge en los últimos años. La educación 3.0³ ya es una realidad, y los nativos digitales⁴ obligarán a cambiar la forma en la que se enseña en la actualidad [1]. Es por esto, que, sabido el auge de las nuevas técnicas de estudio, decidí afrontar el reto y construir por mí mismo una aplicación que cubriese la necesidad existente.

Bien es cierto que podía afrontar la construcción utilizando tecnologías que ya conociese y únicamente mostrar los conocimientos que adquirí durante mi etapa universitaria, esto haría que el resultado final quizás fuese mejor, pero he preferido nuevamente cuestionarme a mí mismo e intentar optar por nuevas tecnologías y herramientas, muchas de ellas desconocidas para mí hasta la fecha. Quizás esta apuesta por innovar y aprender cosas nuevas haga que muchas de las decisiones tomadas y defendidas durante la realización de este proyecto sean cuestionables, pero considero que la motivación es el motor que mueve el mundo, y en este caso, la construcción de una aplicación que cumpliera con las necesidades que usuarios de este tipo de aplicaciones requerimos y que su construcción se realizase empleando

³ **Educación 3.0:** es un término utilizado para describir una variedad de formas de integrar tecnología en el aprendizaje.

⁴ **Nativo Digital:** es un término utilizado para describir a una persona nacida en la era digital.

nuevas técnicas y herramientas, me mantendría motivado a afrontar la realización de este trabajo fin de grado.

1.3 Objetivos

Como objetivo principal se espera cumplir con el diseño y desarrollo de una aplicación híbrida que con la misma base de código permitirá generar versiones para las siguientes plataformas:

- PWA (aplicación web progresiva) [2].
- Aplicaciones móviles (*Android*, *iOS*, ...) a través de *Apache Cordova* [3].
- Aplicaciones de escritorio multiplataforma usando el *framework Electron* [4].

La aplicación resultante estará destinada al aprendizaje de idiomas o pruebas estandarizadas (pruebas de tipo test). El sistema estará diseñado para que los usuarios puedan de una forma sencilla añadir su propio contenido en función de sus necesidades, permitiendo compartir este contenido con el resto de los usuarios de la aplicación si así lo desean. El contenido generado será mostrado a los usuarios de dicha aplicación en forma de *flash cards*, utilizando como método de repetición el algoritmo de *SuperMemo*⁵. Además, la aplicación resultante mostrará al usuario estadísticas del avance en los diferentes cursos en los que participe.

Como objetivos secundarios asociados al trabajo se enumeran los siguientes:

- Garantizar que la aplicación sea fácilmente escalable.
- Garantizar que la aplicación resultante tenga unas condiciones de diseño y usabilidad.
- Diseñar una correcta adaptación de la visualización de los recursos para dispositivos de pantalla reducida.

1.4 Planificación temporal

Para afrontar la realización de este trabajo fin de grado se pretende realizar una división en fases. La *Ilustración 1-1* representa un cronograma donde podemos ver una representación de las fases y su duración aproximada durante las 16 semanas de duración del proyecto. Esta planificación será explicada en detalle en el *capítulo 2*.

Etapas Proyecto	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Cronograma TFG																
1. Etapa inicial																
2. Etapa de elaboración																
3. Etapa de construcción.																
4. Etapa de transición.																

Ilustración 1-1: Cronograma etapas del TFG en semanas

A modo de resumen, he de indicar que las fases principales son las siguientes:

⁵ Más información en: <https://www.supermemo.com/english/algsm11.htm>

1. **Etapa inicial:** aprendizaje y elección de las tecnologías y técnicas empleadas para el desarrollo y documentación del alcance del proyecto.
2. **Etapa de elaboración:** definición de los requisitos y diseño de la arquitectura, instalación y configuración del software y preparación de las plataformas de infraestructura y servicios.
3. **Etapa de construcción:** refinamiento de los requisitos y base de datos, diseño de la interfaz y elaboración de la aplicación.
4. **Etapa de transición:** preparación de la aplicación para su despliegue en producción.

1.5 Tecnologías utilizadas

En este apartado se incluye un listado junto con una breve descripción de las diferentes herramientas y tecnologías empleadas a lo largo del desarrollo de este proyecto, estas herramientas y tecnologías serán explicadas en profundidad en el *capítulo 4*.

- **Git:** software de control de versiones para llevar registro de los cambios en los archivos del proyecto.
- **GitLab:** servicio web de control de versiones y desarrollo de software basado en Git, además proporciona, wiki, sistema de seguimiento de errores y herramientas para realizar integración continua y entrega continua.
- **Balsamiq Mockups:** herramienta para el desarrollo de prototipos de la interfaz gráfica de usuario.
- **Visual Studio Code:** editor de código fuente con control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código
- **Firebase:** plataforma de desarrollo de aplicaciones móviles y web, de entre sus muchos servicios ofertados son utilizados en este proyecto, el servicio de autenticación y la base de datos en tiempo real.
- **Google Lighthouse:** herramienta automatizada para auditoria de aplicaciones.
- **YouTube Data API:** interfaz de programación ofrecida por *Youtube* que permite, entre otras funcionalidades la búsqueda en su plataforma de alojamiento de videos.
- **Pixabay API:** interfaz de programación que permite la búsqueda y recuperación de imágenes gratuitas publicadas en *Pixabay*.

1.6 Estructura del documento

Este documento posee una estructura en capítulos, cada uno de estos capítulos a su vez estarán compuestos de secciones y subsecciones todas ellas numeradas. Detallamos a continuación los capítulos que componen este trabajo junto con una breve introducción resumen del contenido de estos.

Capítulo 1: Introducción. En este capítulo se realiza un repaso general del contexto que conforma este trabajo fin de grado, desde los objetivos del proyecto hasta las herramientas, tecnologías y técnicas empleadas para la realización y planificación.

Capítulo 2: Fases de la realización de trabajo: En este capítulo, realizamos un repaso por la metodología de trabajo empleada y la planificación efectuada para la realización de este trabajo fin de grado.

Capítulo 3: Estado del Arte. En este capítulo se pretende dar al lector una visión sobre las técnicas de repaso espaciado, qué son, qué ventajas aportan y qué aplicaciones existen actualmente o han existido en el mercado (historia de la evolución tecnológica). Por último, dentro de este capítulo realizaremos una crítica sobre las problemáticas detectadas en las soluciones existentes que han dado lugar a la realización de la aplicación resultante.

Capítulo 4: Tecnologías y Herramientas utilizadas. En este capítulo se detallarán las herramientas, técnicas y tecnologías empleadas para la realización de este trabajo, realizando de las que consideramos más influyentes en el proyecto una comparativa del motivo de su elección frente a otras.

Capítulo 5: Análisis del sistema. En este capítulo se realizará un análisis de requisitos que determine cuál será la solución resultante realizando una definición formal del sistema y que requisitos debe cumplir.

Capítulo 6: Diseño y construcción del sistema. En este capítulo se explicará la arquitectura del sistema, el modelo de datos y el diseño de interfaces de usuario de la aplicación, además se realizará un repaso por el modo de construcción, analizando las diferentes técnicas empleadas y cómo se han utilizado las herramientas y tecnologías descritas en los capítulos anteriores.

Capítulo 7: Pruebas y control de calidad del sistema. En este capítulo se plantea un plan de pruebas que permita validar los casos de uso definidos. Además, se presentan los resultados de un control de calidad aportando diferentes métricas.

Capítulo 8: Conclusiones y trabajo futuro. En este capítulo se realizará un repaso por los objetivos planteados, detallando cuales de ellos han sido alcanzados y cuáles no. Además, se plantearán una serie de líneas de desarrollo futuro, ampliaciones y mejoras que no han sido acabadas o planteadas en la realización de este TFG.



Capítulo

2

FASES DE LA REALIZACIÓN DEL TRABAJO

2.1 Introducción

En el día a día en la realización de un proyecto, se hace necesario el uso de una metodología y una planificación del trabajo que garantice el cumplimiento de los objetivos y la **gestión eficaz del proyecto**. Esto es esencial para culminar el proyecto con éxito. Dicha planificación es lo que se conoce como la **gestión del proyecto**, y permite la elaboración de planificaciones, manteniendo controlados todos los aspectos relevantes de un proyecto de una forma homogénea [5].

En este capítulo se realiza un repaso sobre los aspectos relacionados con la planificación y la organización temporal del proyecto, realizándose además una introducción sobre los conceptos propios de la metodología de trabajo empleada, justificándose su elección y las variaciones efectuadas y razonándose el motivo por el que fueron realizadas.

2.2 Metodología de trabajo

2.2.1 Elección de la metodología de trabajo

Aplicar una metodología de desarrollo software de las muchas existentes, centrándose en cumplir todos sus principios y normas, considero que es algo difícil para la elaboración de este trabajo fin de grado, esto es debido a que las metodologías consultadas se centran en estructurar, planear y controlar el proceso de desarrollo en sistemas en los que existe un equipo de trabajo y un cliente. En la realización de este trabajo fin de grado no existe un cliente como tal y el equipo de trabajo está conformado por un único integrante. Además, algunas metodologías recomiendan el uso de herramientas (comunicativas, de diseño...), que considero no aportan valor a la realización de este proyecto por su envergadura y limitaciones. Es por este motivo, por el que, si afirmase que al realizar este trabajo estoy aplicando una metodología de desarrollo específica, sería incoherente, ya que quedarían muchos aspectos por cumplir.

Sabido lo expuesto con anterioridad, es por lo cual decidí buscar que metodología de las existentes se adaptaba mejor a la realización de este trabajo. Desde un inicio, tenía claro que aplicar un enfoque **iterativo e incremental** frente a un método en cascada tradicional tendría sus ventajas, ya que permitiría centrarse en la evolución de prototipos ejecutables que permitan la validación de funcionalidades y requisitos. Esta evolución de los prototipos garantizará que a la finalización de este trabajo se dispondrá de un entregable de la aplicación, que quizás no cumple todos los requisitos propuestos, pero al menos cumplirá aquellos que aportan mayor valor a la aplicación.

Entre una de las metodologías más importantes e influyentes iterativas e incrementales se encuentra **Proceso Racional Unificado** [6] (de ahora en adelante, RUP), el cual hemos decidido seleccionar como metodología de referencia para la realización de este trabajo fin de grado, su elección viene condicionada principalmente por los siguientes motivos:

- Utiliza una **arquitectura basada en componentes**⁶: como veremos en el *capítulo 6.3* de este trabajo, la aplicación está conformada por un conjunto de componentes.
- No es una metodología con pasos firmemente establecidos, sino un conjunto de metodologías **adaptables al contexto** y necesidades de cada propósito, lo que nos permite emplearla dadas las restricciones de este proyecto.
- Es una implementación del **desarrollo en espiral**⁷, donde el ciclo de vida organiza las tareas en **cuatro grandes fases** bien definidas, cada una de estas fases son divididas en iteraciones que dan como resultado un ejecutable. Esto garantiza que, a la finalización de cada una de las

⁶ **Componente Software**: Un componente se puede definir como una pieza de software no trivial: un módulo, paquete, o un subsistema que cumple una funcionalidad clara, tiene un límite claro y puede integrarse en una arquitectura bien definida [52]

⁷ **Desarrollo en espiral**: es un modelo de ciclo de vida del software se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades que no están fijadas a ninguna prioridad, sino que se eligen en función del análisis de riesgo.

cuatro fases principales existirá al menos un entregable para este trabajo fin de grado.

2.2.2 Introducción al Proceso Unificado Rational - RUP

En una variedad de libros se define RUP como un **proceso de ingeniería del software bien definido y bien estructurado** con un enfoque de desarrollo software iterativo, centrado en la arquitectura e impulsado por casos de uso [6] [7].

La forma más sencilla de tener una visión clara del enfoque de trabajo de esta metodología es el *RUP hump chart* que podemos ver en la *Ilustración 2-1*. En este gráfico podemos apreciar que a la izquierda se encuentran agrupadas las principales actividades dentro de un proyecto de desarrollo software. Referente a la parte superior del gráfico, se encuentra dividido en un total de cuatro fases (Inicio, Elaboración, Construcción y Transición), estas son las fases que atravesará nuestro proyecto durante su ciclo de vida y de las que realizaremos un repaso por su aplicación al proyecto en la *sección 2.3*. de este capítulo.

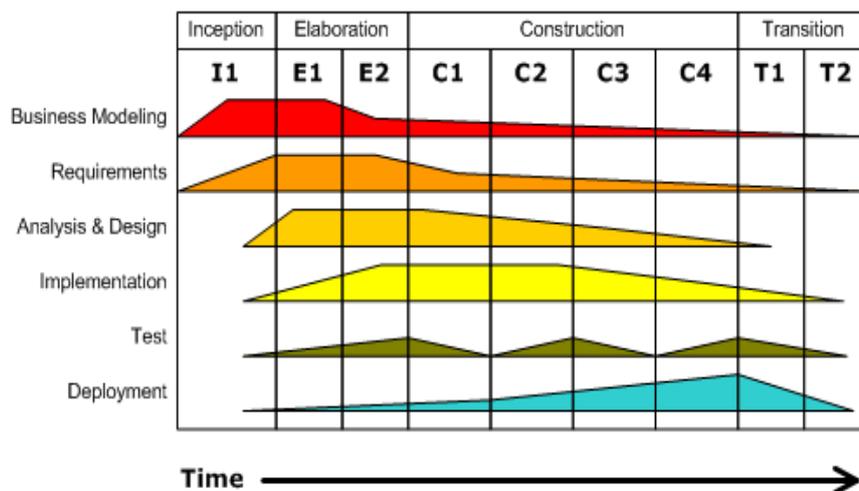


Ilustración 2-1 Gráfico Hump metodología RUP

Las primeras iteraciones de la fase de inicio y elaboración se enfocan en la comprensión del problema, la tecnología y el diseño de una arquitectura, haciendo mayor énfasis en las actividades del modelado de negocio y de requisitos. Durante las iteraciones de la fase de elaboración, que supone gran parte del total del proyecto, se enfocan las iteraciones al refinamiento del análisis y diseño e implementación. Por último, en la fase de transición se pretende hacer accesible y entendible el producto para su entrega.

2.3 Planificación del proyecto

2.3.1 Planificación de las etapas

Atendiendo a la metodología RUP brevemente descrita en el apartado 2.2.2 de este capítulo, el proyecto se ha estructurado en un total de 4 fases (Inicio, Elaboración, Construcción y Transición). En la *Tabla 2-1* podemos ver claramente la planificación establecida inicialmente con el número de horas total asignadas a cada etapa, la fecha prevista de comienzo y la de fin.

Fase	N.º Iteraciones	H. Previstas	Fecha inicial	Fecha Final	Días
Inicial	1	54	21/05/2018	07/06/2018	18
Elaboración	2	78	08/06/2018	03/7/2018	26
Construcción	7	174	04/07/2018	30/08/2018	58
Transición	1	30	31/08/2018	10/09/2018	10
TOTAL	4	336			112

Tabla 2-1: Planificación de fases del proyecto

Como podemos apreciar en la *Tabla 2-1*Tabla 2-1: Planificación de fases del proyecto, este fue planificado con fecha de comienzo 21/05/2018 y fecha de fin 10/09/2018 constando el mismo de un total de 112 días lo que conforma un total de 16 semanas de duración. Para conseguir esta planificación se ha creado un recurso en *Microsoft Project* al cual se estableció una jornada laboral de 21 horas semanales durante las 16 semanas de duración del proyecto, esto supone un total de 3 horas diarias considerando todos los días entre la fecha de inicio del proyecto y el fin como días laborables. Esta jornada laboral se encuentra definida en *Microsoft Project* como podemos apreciar en la *Ilustración 2-2*.

Leyenda:

- Laborable
- No laborable
- 31** Horas laborables modificadas

En este calendario:

- 31** Día de excepción
- 31** Semana laboral no predeterminada

Haga clic en un día para ver sus periodos laborables: Periodos laborables del 21 mayo 2018:

• 18:00 a 21:00

Basado en:
Excepción 'Horario Mayo' en el calendario 'Estándar'.

Nombre	Comienzo	Fin
1 Horario Mayo	21/05/2018	31/05/2018
2 Horario Junio	01/06/2018	30/06/2018
3 Horario Julio	02/07/2018	31/07/2018
4 Horario Agosto	01/08/2018	31/08/2018
5 Horario Septiembre	01/09/2018	09/09/2018

Ilustración 2-2 Calendario del recurso Juan asignado al proyecto

2.3.2 Etapa inicial

Al tratarse de un proyecto de una envergadura reducida, esta etapa se divide en una única iteración en la que se realizaron tareas relacionadas con la comprensión de lo que se pretende construir, se identificaron las funcionalidades clave del sistema y se tomaron decisiones sobre el proceso y las herramientas que han sido empleadas con posterioridad.

Las tareas de la única iteración existente se encuentran definidas en *Microsoft Project* siendo su diagrama de Gantt con la planificación y programación la que podemos ver en la *Tabla 2-2*.

Id	Nombre de tarea	Duración	Comienzo	junio 2018						
				21	24	27	30	02	05	08
2	#F1 - FASE INICIAL	54 hrs	lun 21/05/18	[Barra de 54 hrs]						
3	Iteración I1	54 hrs	lun 21/05/18	[Barra de 54 hrs] Iteración						
4	Definición del ambito y objetivos del proyecto	5 hrs	lun 21/05/18	[Barra de 5 hrs]						
5	Estudios diferentes técnicas de repaso espaciado	2 hrs	mié 23/05/18	[Barra de 2 hrs]						
6	Estudio de los conceptos basicos de Vue.js	6 hrs	jue 24/05/18	[Barra de 6 hrs]						
7	Elección del entorno de trabajo.	9 hrs	sáb 26/05/18	[Barra de 9 hrs]						
8	Generar documento de visión.	8 hrs	mar 29/05/18	[Barra de 8 hrs]						
9	Definir cronograma del proyecto	3 hrs	mié 30/05/18	[Barra de 3 hrs]						
10	Definir las capacidades y objetivos del proyecto	4 hrs	vie 01/06/18	[Barra de 4 hrs]						
11	Estudios previos, aplicaciones de aprendizaje.	6 hrs	sáb 02/06/18	[Barra de 6 hrs]						
12	Documentación de la iteración 1 en el documento TFG	9 hrs	mar 05/06/18	[Barra de 9 hrs]						

Tabla 2-2: Iteraciones y tareas de la fase inicial

En esta iteración haciendo caso de las recomendaciones de la metodología RUP [6] se definió un **documento de visión** en el que se definieron los objetivos, las principales tecnologías que han sido empleadas, los beneficios, las oportunidades, el problema que resuelve y que es lo que realizara la aplicación resultante. Todo esto descrito a un nivel muy alto. Este documento de visión marcó unas bases claras para la realización de este proyecto y dio lugar a la entrega con fecha *30/06/2018* del **documento del anteproyecto** de este trabajo fin de grado.

2.3.3 Etapa de Elaboración

En esta etapa han sido definidas un total de **dos iteraciones**, una primera iteración cuyo fin es el refinamiento de los objetivos del proyecto, definiendo unos **requisitos estables** y diseñando una arquitectura sólida. Además, fue

diseñado el primer prototipo ejecutable de la aplicación (mockups) que demuestra el comportamiento y diseño de las partes principales. El desarrollo de esta primera iteración dio lugar a los artefactos que han sido plasmados en el *capítulo 5* de este trabajo.

Por último, esta etapa consta de una segunda iteración cuyo objetivo es el de **establecer el sistema de gestión de configuración del software (GCS/SCM)** ⁸ adecuado para el desarrollo de la aplicación y que describiremos en detalle en el complemento trabajo fin de grado que acompaña a este trabajo.

Cada una de las tareas de esta iteración, se encuentran definidas en *Microsoft Project* siendo su diagrama de Gantt con la planificación y programación la que podemos ver en la *Tabla 2-3*.

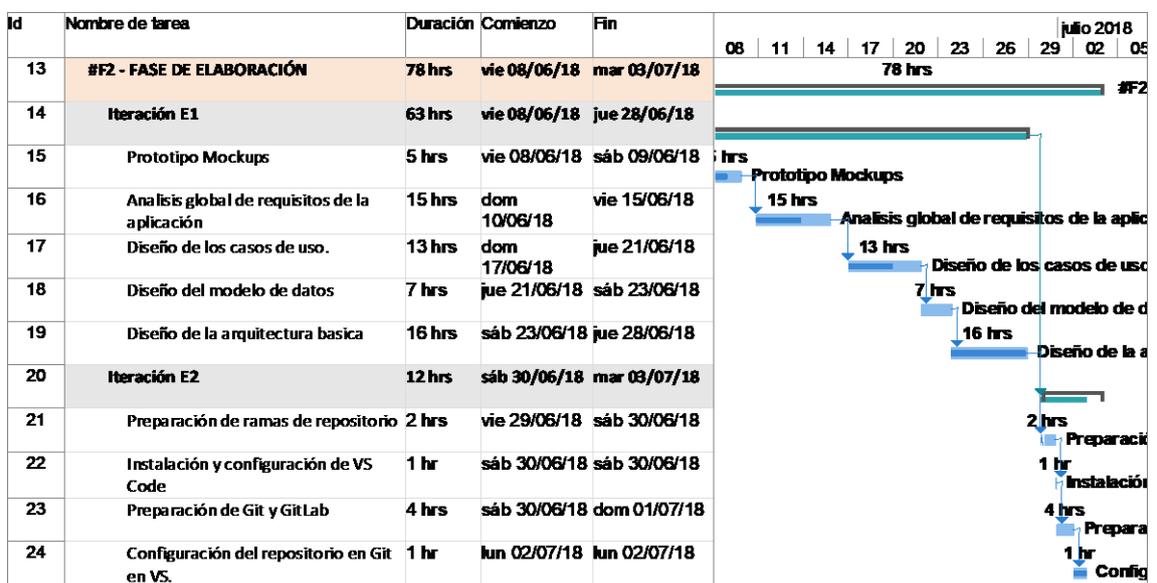


Tabla 2-3: Iteraciones y tareas de la fase de elaboración

2.3.4 Etapa de Construcción

En la etapa de construcción los objetivos fueron centrados en el diseño detallado y ajuste de los requisitos de cada funcionalidad de la aplicación, su implementación y las pruebas necesarias para garantizar su correcto funcionamiento y el del sistema completo. Esta es la etapa para la que más horas de trabajo se han preestablecido en la planificación con un total de 174 de las 336 horas previstas, esto supone un **51% del trabajo total**.

⁸ Sistema de gestión de configuración del software (GCS/SCM): es el conjunto de procesos destinados a asegurar la calidad de todo producto, a través del estricto control de los cambios realizados sobre los mismos y de la disponibilidad constante de una versión estable de cada elemento.

Las tareas de las iteraciones se encuentran definidas en Microsoft Project siendo su diagrama de Gantt con la planificación y programación la que podemos ver en las tablas: *Tabla 2-4*, *Tabla 2-5* y *Tabla 2-6*

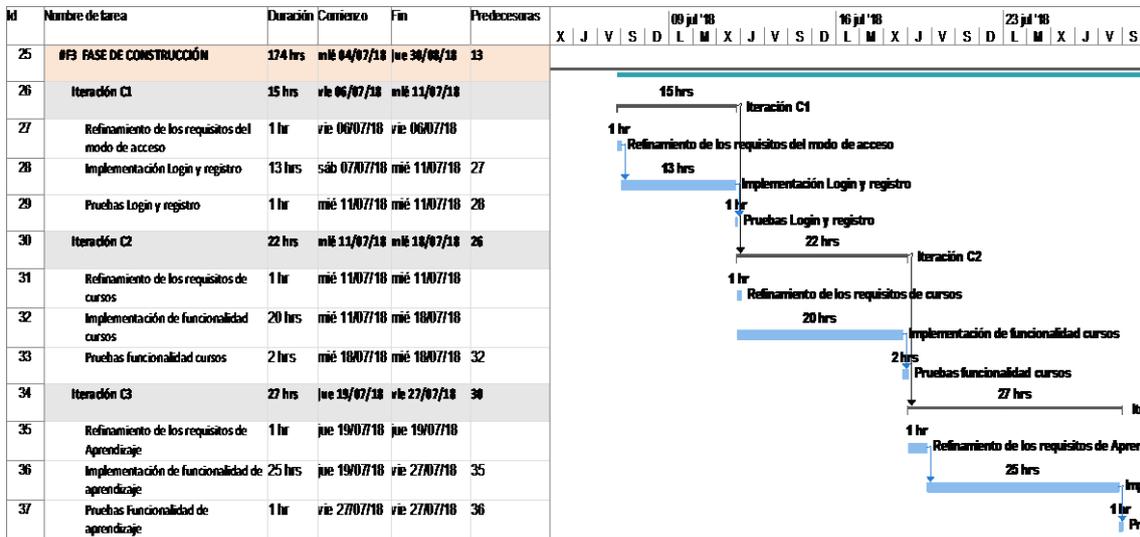


Tabla 2-4: Iteraciones C1-C3 fase de construcción



Tabla 2-5 Iteraciones C4-C6 fase de construcción

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	13 ago '18							20 ago '18							27 ago '18						
						S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V
48	Iteración C7	16 hrs	sáb 11/08/18	jue 16/08/18	47																					
49	Diseño de tarjetas aprendizaje de idiomas	8 hrs	sáb 11/08/18	lun 13/08/18																						
50	Implementación de tarjetas de idiomas	8 hrs	mar 14/08/18	jue 16/08/18	49																					
51	Iteración C8	16 hrs	jue 16/08/18	mar 21/08/18	48																					
52	Diseño de tarjetas tipo test	8 hrs	jue 16/08/18	dom 19/08/18																						
53	Implementación de tarjetas tipo test	8 hrs	dom 19/08/18	mar 21/08/18	52																					

Tabla 2-6: Iteraciones C7-C8 fase de construcción

Como podemos apreciar en las tablas referenciadas con anterioridad, existen un total de 8 iteraciones, esta cantidad se encuentra en el límite de las recomendaciones de RUP que aconseja un número de iteraciones generalmente entre dos a cuatro veces más que el resto de las fases del proyecto [6]. En nuestro caso hemos preferido optar por el número mayor de iteraciones y acotar cada una de ellas como una funcionalidad de nuestra aplicación, siendo ordenadas en un orden decreciente atendiendo a los siguientes criterios:

1. Funcionalidades principales del sistema necesarias para el desarrollo de otras.
2. Funcionalidades con un riesgo alto debido al desconocimiento de la tecnología empleada.
3. Funcionalidades que aportan mayor valor a la aplicación.

El acotar las iteraciones a una funcionalidad concreta viene justificado por la idea de que en la fase de construcción trabajamos con tecnologías y herramientas desconocidas hasta el comienzo de este trabajo, por lo que una estimación en la duración de cada iteración es difícil de realizar. Afrontando las tareas atendiendo a la ordenación descrita, garantizamos que, al fin de la planificación efectuada para la fase de construcción, es posible que no estén completadas todas las iteraciones planificadas, pero si lo estarán las más influyentes e importantes para el proyecto. Por lo tanto, tendremos una versión de la aplicación funcional con las principales características implementadas. De igual modo, si al alcanzar la fecha fin de la fase las iteraciones se han completado dentro de la planificación prevista, es posible se añadan más funcionalidades que aporten un valor añadido al proyecto.

Por último, hay que indicar que durante toda esta fase se hace uso del **sistema de gestión de configuración del software (GCS/SCM)** el cual fue creado en la fase de elaboración y que ha permitido atendiendo a las recomendaciones de la metodología RUP, integrar cada una de las iteraciones probando continuamente el sistema [6], esto es posible gracias a la puesta en marcha de un procedimiento de CI/CD⁹. Veremos cómo se ha realizado

⁹ **CI/CD**: Estas siglas se refieren al proceso de integración continua y entrega continua. Siendo esta la práctica de fusionar todas las copias de trabajo de los desarrolladores permitiendo construir, probar y lanzar software con mayor velocidad y frecuencia posible.

esto en detalle en el complemento al trabajo fin de grado que acompaña a este trabajo.

2.3.5 Etapa de Transición

Por último, llegamos a la fase de transición, esta fase se encuentra dividida únicamente en una iteración que se centrará en asegurarse que la **versión beta** totalmente funcional producida en la fase de construcción cumpla lo esperado. Además, se corregirán pequeños defectos que puedan ser detectados y se preparara la aplicación para su despliegue en producción.

Las tareas de las iteraciones se encuentran definidas en *Microsoft Project* siendo su diagrama de Gantt con la planificación y programación la que podemos ver en la *Tabla 2-7*.

Id	Nombre de tarea	Duración	Comienzo	Fin	03 sep '18							10 sep '18						
					V	S	D	L	M	X	J	V	S	D	L	M		
54	#F4 FASE DE TRANSICIÓN	30 hrs	vie 31/08/18	dom 09/09/18	30 hrs													
55	Iteración T1	30 hrs	vie 31/08/18	dom 09/09/18	30 hrs													
56	Pruebas de integración	5 hrs	vie 31/08/18	sáb 01/09/18	5 hrs													
57	Análisis de auditoría del sistema	4 hrs	sáb 01/09/18	dom 02/09/18	4 hrs													
58	Preparación del sistema para CI/CD para el despliegue en producción	1 hr	lun 03/09/18	lun 03/09/18	1 hr													
59	Manual de usuario	4 hrs	lun 03/09/18	mar 04/09/18	4 hrs													
60	Mejoras en aspectos de diseño	7 hrs	mar 04/09/18	jue 06/09/18	7 hrs													
61	Corrección de errores	9 hrs	vie 07/09/18	dom 09/09/18	9 hrs													

Tabla 2-7: Iteraciones y tareas de la fase de transición

Cabe destacar que RUP difiere radicalmente del desarrollo tradicional, principalmente porque ingresa a la fase con una versión del sistema razonablemente estable, integrada y probada durante la fase de construcción [6].

2.4 Seguimiento del proyecto.

Como podemos apreciar en la *Ilustración 2-3* obtenida de los gráficos aportados en el repositorio de *Gitlab*, se han realizado un total de 218 integraciones de código en la rama *develop* del repositorio entre los periodos de junio a septiembre de 2018.

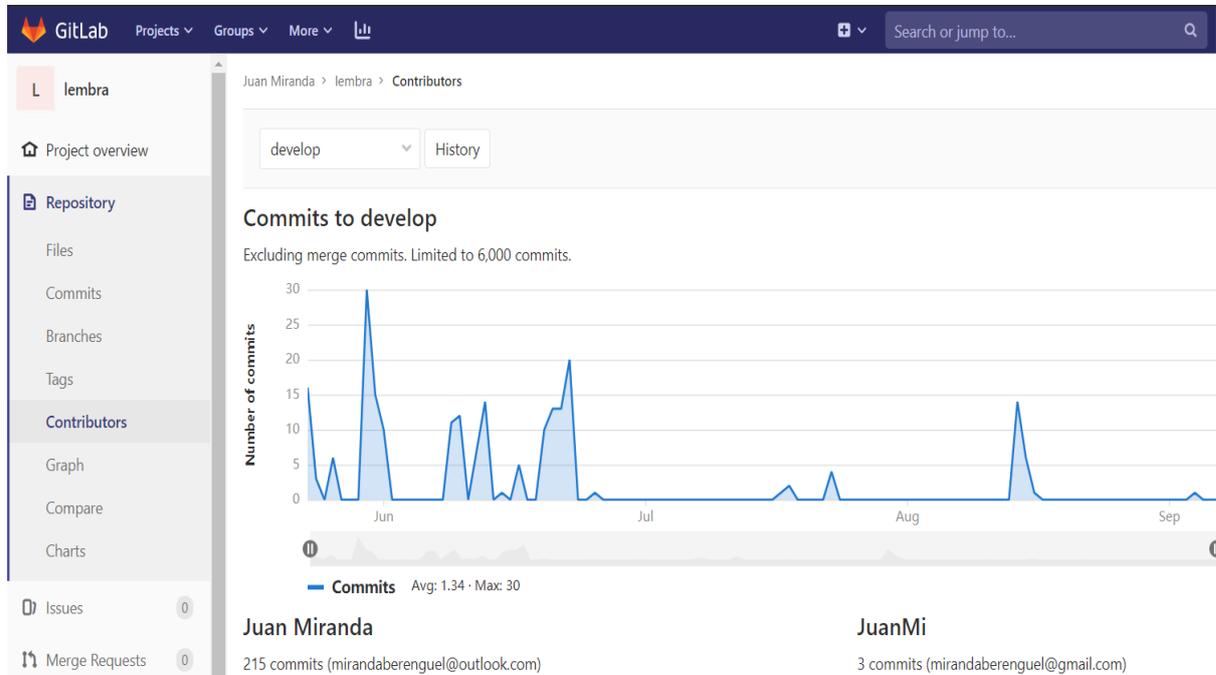


Ilustración 2-3: Grafica de commits en repositorio

Gitlab nos aporta más estadísticas en forma de grafico sobre cuándo se han realizado las integraciones de código en el repositorio. En la *ilustración Ilustración 2-4* y *Ilustración 2-5* podemos ver el número de *commit*¹⁰ realizados por día de la semana o las horas en las que se han realizado el mayor número de *commits*

¹⁰ **Commits:** confirmar un conjunto de cambios provisionales de forma permanente en el repositorio de código fuente de la aplicación.

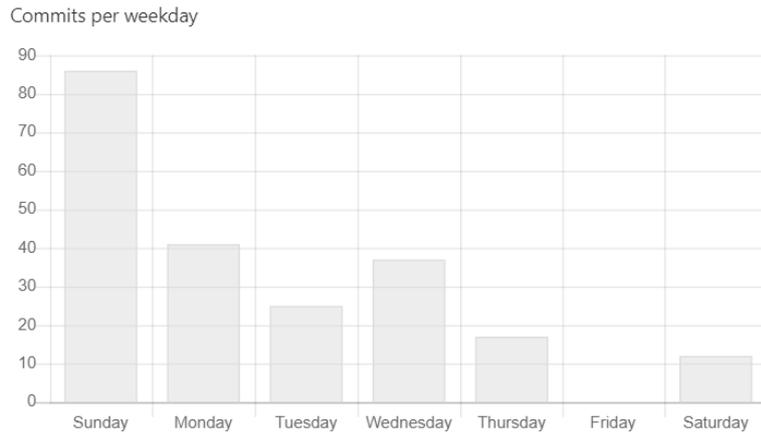


Ilustración 2-4: Commits por día de la semana.

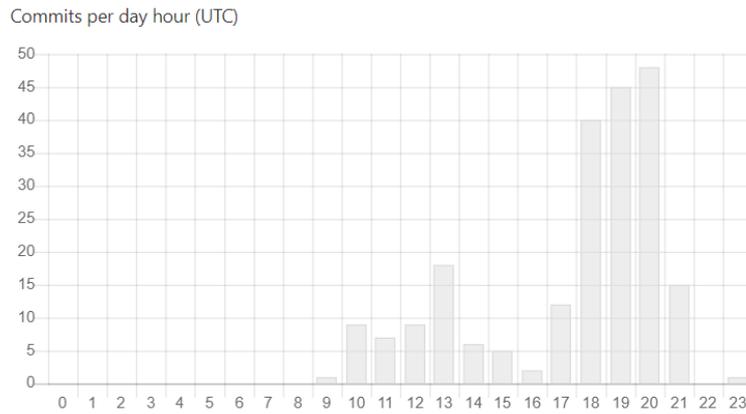


Ilustración 2-5: Relación de Commits por franja horaria.



Capítulo

3

ESTADO DEL ARTE

3.1 Introducción

En 2007 un video denominado *A Visión Of Students Today* [8] realizado por 200 estudiantes de la *University of Kansas*, hicieron un intercambio de ideas de como los estudiantes aprenden, lo que necesitan para aprender y cómo encajaba el sistema educativo existente. En este video comparan cuestiones como, las dos horas que pasan con el móvil diarias, frente las tres que asisten a clase. Dando una visión sobre los métodos educativos empleados, y la necesidad de un cambio en los métodos de aprendizaje existentes.

Gracias a internet y la cultura de lo abierto y lo público, han surgido nuevas formas de aprendizaje [9] [10]. El sistema educativo formal, están en constante cambio, hace unos años era impensable que un profesor compartiese videos educativos en plataformas como *YouTube*, hoy esto es una realidad y universidades como la *UPV - Universidad Politécnica de Valencia* comparten material impartido por su profesorado, consiguiendo desde el año 2010 más de 36 millones de visualizaciones¹¹.

En este capítulo realizaremos un análisis de las nuevas formas educativas y cuáles son los intereses de los estudiantes en la era de la educación digital. Además, realizaremos un repaso de las aplicaciones actuales en el mercado y que realizan funcionalidades iguales o parecidas a las que se pretende desarrollar en este trabajo fin de grado.

Por último, atendiendo a las conclusiones obtenidas del análisis efectuado, realizaremos una crítica de lo existente en la actualidad y que es lo que la aplicación resultante de este trabajo fin de grado pretende cumplir para suplir la necesidad existente.

¹¹ Fuente: Canal UPV YouTube, Fecha de consulta: 29/08/2018, visualizaciones totales: 36.433.835

3.2 La curva del olvido y el repaso espaciado

En 1885 *Hermann Ebbinghaus*¹² realizó el hallazgo de lo que hoy conocemos como **curva del olvido**, que indica cuánto se mantiene un contenido en el cerebro [11]. Para evitar el olvido, existen diferentes técnicas de aprendizaje, una de estas técnicas de aprendizaje es el **repaso espaciado** en la que se repasa a intervalos crecientes. Es decir, para empezar el sujeto se familiariza con la información para fijarla en su memoria, posteriormente se realizan repasos periódicos y, conforme se va consolidando la información en la memoria, los repasos se distanciarán más y más en el tiempo. Una representación de estos conceptos es el gráfico de la *Ilustración 3-1* donde, vemos representada la curva del olvido que sigue un decrecimiento exponencial y como tras varios repasos espaciados durante varios días esta curvatura decrece.

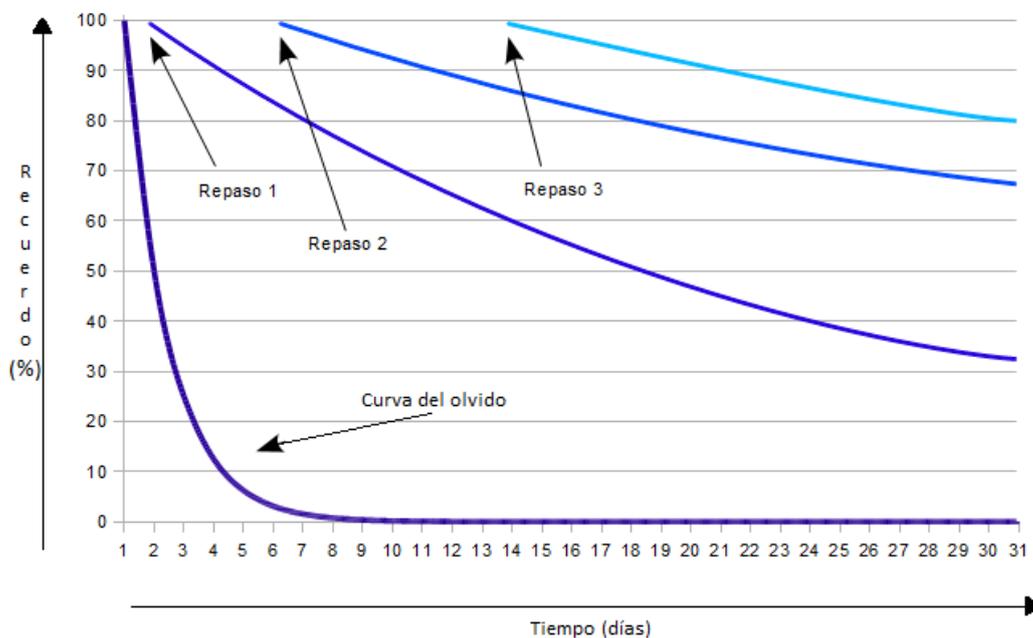


Ilustración 3-1: Curva del olvido Ebbinghaus

Un gran número de investigaciones de la psicología cognitiva y educativa han demostrado que la **práctica espaciada** descrita es una forma de mejorar la **efectividad y la eficiencia del aprendizaje** [12] [13]. Además, esta tiene un enorme potencial para mejorar los resultados educativos [14]. Una vez que sabemos esto, la cuestión que debemos plantearnos es si realmente existe un momento óptimo para repasar lo aprendido. Según *Woźniak* sí existe, lo dedujo en 1987 cuando se dio cuenta de que las computadoras podían calcular el momento del olvido. El resultado de su investigación dio lugar a la aparición del algoritmo *SuperMemo*, el cual según sus afirmaciones permite determinar el momento óptimo para repasar lo aprendido [15].

Estudios posteriores han deducido que las planificaciones para repasar lo aprendido dependerán del tipo de contenido y del tiempo que pretendemos

¹² **Hermann Ebbinghaus**: Filósofo y psicólogo alemán que publicó estudios referentes al olvido en función del tiempo, relación también conocida con el nombre de curva del olvido.

que perdure en nuestras mentes. Es decir, no existe el espaciado óptimo aplicable de forma genérica a todo lo que pretendemos aprender. Esto se deduce de las diferentes pruebas realizadas en la revista *Experimental Psychology*, en 2009 [16].

3.2.1 Técnicas de repaso espaciado

3.2.1.1 Sistema de Leitner

El *sistema de Leitner* [17] [18] es la implementación más simple del principio de repetición espaciada. En este método, las tarjetas de aprendizaje son clasificadas en función al grado de conocimiento de cada una de ellas. Para comprender su funcionamiento podemos apreciar la *Ilustración 3-2* donde las tarjetas de aprendizaje están clasificadas en grupos, en función de si se recuerda la respuesta a la pregunta, la tarjeta avanza al siguiente grupo. Cada grupo sucesivo tiene un período de tiempo más largo antes de que el alumno tenga que volver a repasar las tarjetas

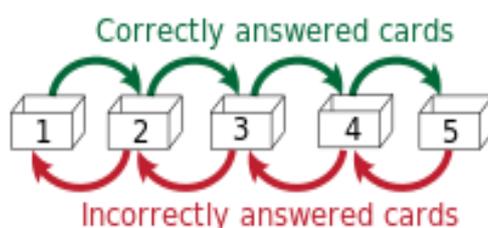


Ilustración 3-2: Sistema de Leitner

Este sistema de repeticiones fue propuesto en 1970 por el comentarista y divulgador de la ciencia *Sebastian Leitner*. Su publicación se realizó en el libro *So lernt man lernen* (Cómo aprender a aprender), el cual es un manual práctico sobre la psicología del aprendizaje y que se convirtió en un *best-seller*¹³.

3.2.1.2 Algoritmo SuperMemo

Formulado por *Woźniak*, en 1985, en una versión en papel, este algoritmo fue denominado SM0 [15], donde el último número representa una extensión que indica la versión actual del algoritmo. En la actualidad el algoritmo *SuperMemo* se encuentra en su versión 17 publicada el 11 de junio de 2018. Esta versión, al igual que muchas de sus antecesoras, no son públicas y forman parte de la actividad comercial de la empresa *SuperMemo World* [19].

Durante los 33 años de existencia a la fecha de realización de este trabajo, la formulación inicial del algoritmo de repetición espaciada ha aumentado su eficiencia, pero a su vez su complejidad. [20]. El algoritmo SM2 es

¹³ **Best-seller:** Obra literaria escrita sobre un tema que capta fácilmente la atención del lector, con estilo y vocabulario sencillos y buscando fundamentalmente convertirse en un gran éxito de ventas

considerado como la versión más popular, esto es debido a su simplicidad frente a las versiones posteriores. SM2 utiliza el rendimiento de una tarjeta para programar la próxima repetición solo de esa tarjeta, en cambio las versiones posteriores del algoritmo utilizan el rendimiento de una tarjeta para programar la próxima repetición de esa tarjeta y otras similares [21].

3.3 Tecnologías digitales de aprendizaje

Es importante, previo a la construcción de una aplicación para el aprendizaje, conocer que opinan los usuarios sobre este tipo de aplicaciones. Cuestiones como: ¿Qué tipo de dispositivos utilizan?, ¿qué tipo de tecnologías prefieren?, o si, ¿consideran útil emplear tecnología en el aprendizaje? Son consideraciones necesarias y útiles antes de afrontar la construcción de una aplicación. Para dar respuesta las preguntas propuestas, en los consiguientes puntos de este apartado intentaremos dar una visión sobre la opinión de los usuarios y expertos [22] [23] [24].

3.3.1 Efecto de la tecnología de aprendizaje digital

Existen defensores y detractores del uso de **tecnologías digitales de aprendizaje** (de ahora en adelante, DLT), pero la realidad es que en 2016 el 79% de los estudiantes universitarios cree que la DLT impactó positivamente en sus calificaciones. Estos datos [25] son extraídos del gráfico de la *Ilustración 3-3*.

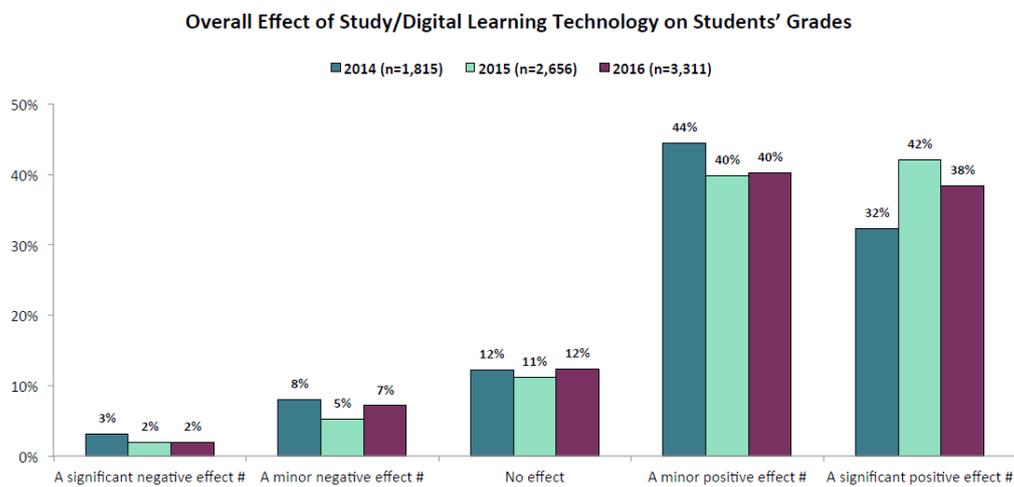


Ilustración 3-3: Efecto de las DLT en las calificaciones de los estudiantes [25]

La importancia y el interés por las DTL están ocasionando la proliferación en los últimos años de aplicaciones orientadas a la educación y sobre las que realizamos un repaso en la *sección 3.4* de este capítulo. Algunas de estas

aplicaciones hacen uso de lo que se conoce como **aprendizaje adaptativo**¹⁴, siendo este un método educativo que emplea, entre otras técnicas, **repetición espaciada**.

Según McGraw-Hill Education editorial estadounidense que ofrece contenido enfocado a la educación, durante los últimos años ha pasado de un modelo comercial basado en la impresión a uno basado en la entrega de contenido digital y soluciones DTL. La editorial mencionada realizó un estudio sobre la opinión del aprendizaje adaptativo y su utilidad en la retención de contenidos [25]. Los resultados, que podemos ver en el gráfico de la *Ilustración 3-4*, indican que el 67% de los encuestados opina que su uso es útil o muy útil.

Perceived Helpfulness of Online Quizzes and Adaptive Learning Technology for Retention

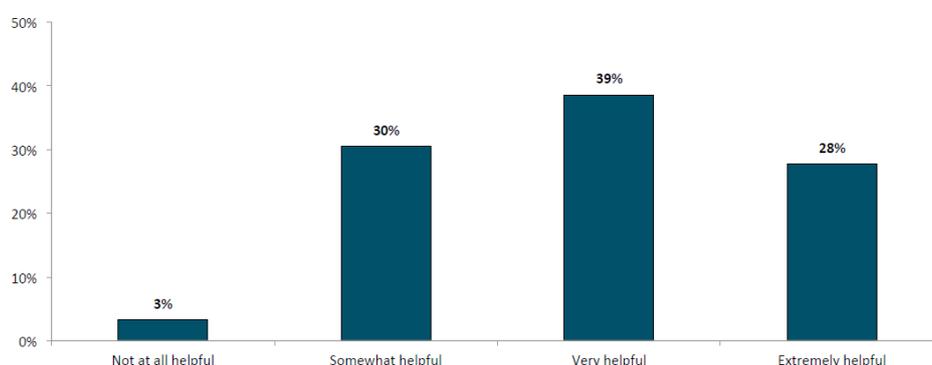


Ilustración 3-4: Utilidad de la tecnología de aprendizaje adaptativa para la retención de nuevos conceptos [25]

3.3.2 Dispositivos electrónicos utilizados para estudiar

Existen diferentes tipos de dispositivos electrónicos desde los que es posible estudiar. Dispositivos como los teléfonos móviles pueden tener ventajas como la movilidad frente a otros como los ordenadores de sobremesa. Pero también tienen desventajas como puede ser el tamaño de pantalla reducido, lo que obliga a un contenido adaptado a los mismos. Es por esto que los usuarios tienen sus preferencias a la hora de seleccionar el tipo de dispositivo desde el que estudiar, indicativo qué debemos tener en cuenta ante el lanzamiento de una aplicación destinada a la educación, siendo de gran importancia conocer las preferencias de los usuarios.

¹⁴ **Aprendizaje adaptativo:** Es un método educativo que utiliza dispositivos de enseñanza interactiva adaptado a la presentación de material educativo de acuerdo con las necesidades de aprendizaje de los estudiantes.

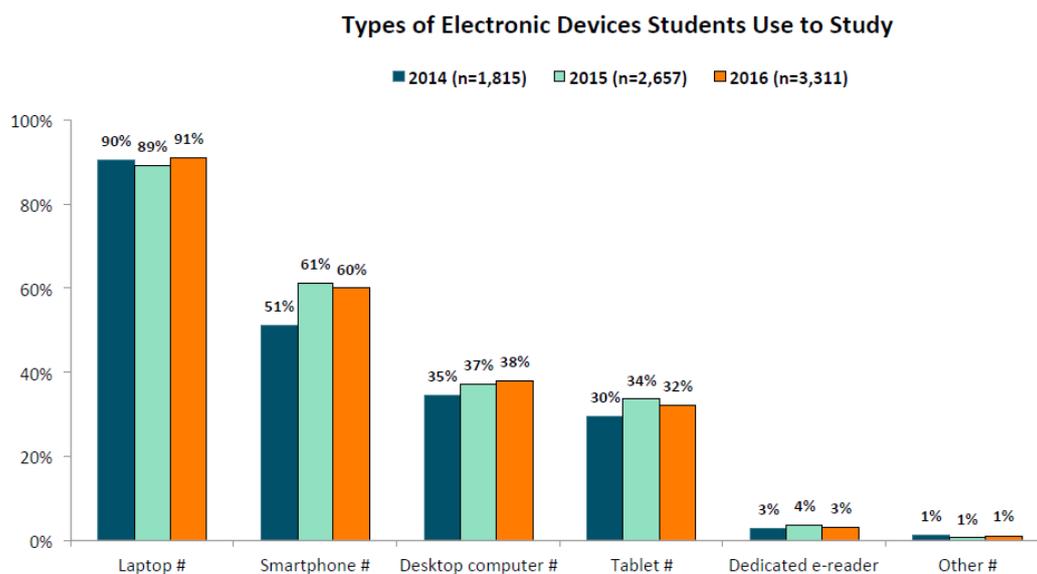


Ilustración 3-5: Tipos de Dispositivos Electrónicos que Usan los Estudiantes para Estudiar [25]

Como podemos apreciar en el gráfico de la *Ilustración 3-5* el uso del ordenador portátil como medio de estudio es preferido por la mayoría de los estudiantes encuestados. Aun siendo significativo el éxito del ordenador por parte de los usuarios como método preferido de estudio, debemos apreciar que el uso de *Tablet* y *Smartphone* es el favorito por entorno al 58% y 36% respectivamente, porcentajes muy a tener en cuenta a la hora de seleccionar el dispositivo para el que ofrecer una aplicación orientada a la educación. Por lo tanto, organizar el contenido para ofrecer una experiencia de usuario óptima en varios tamaños de pantalla es un reto para esta aplicación.

3.3.3 Tipos de tecnología digital que utilizan los estudiantes

Entre las aplicaciones que analizaremos en el apartado 3.4, existen aplicaciones que emplean únicamente la repetición de preguntas prácticas, con un orden preestablecido sin adaptarse al rendimiento del usuario. Otras por el contrario se adaptan al aprendizaje del usuario y emplean video y audio para mejorar el aprendizaje.

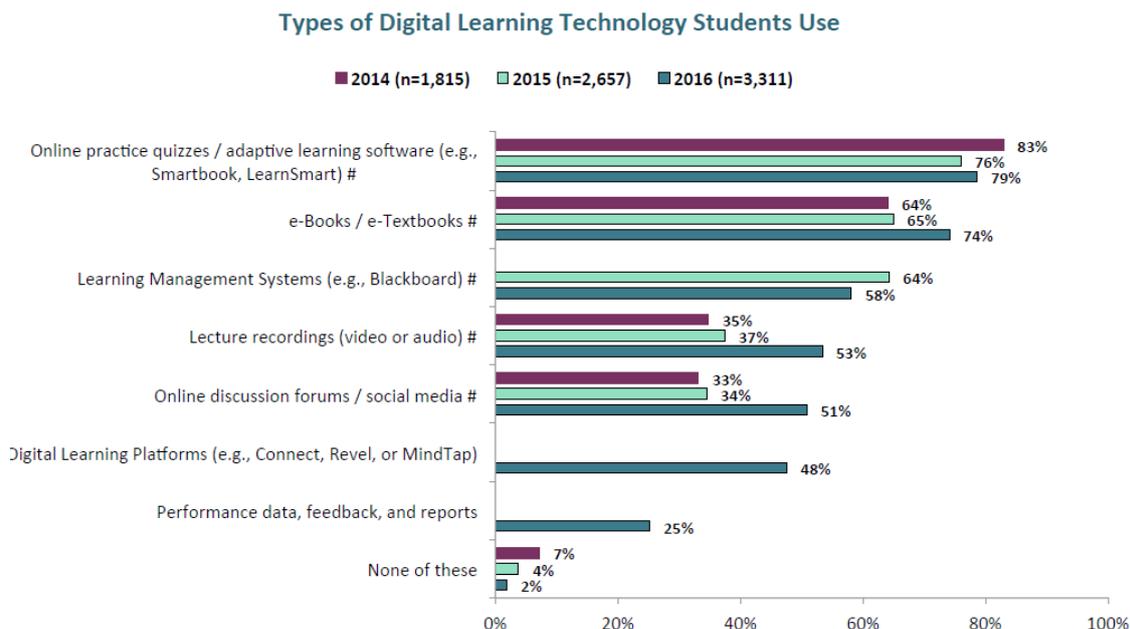


Ilustración 3-6: Tipos de tecnología digital de aprendizaje que utilizan los estudiantes [25]

Como podemos apreciar en el gráfico de la *Ilustración 3-6* existe un mayor interés en el uso **software de aprendizaje adaptativo y pruebas tipo test** entre los estudiantes. Hay que destacar también la fuerte crecida obtenida por el uso de video y audio por el que tienen interés en torno al 50% de los encuestados el último año.

3.3.4 Impacto del análisis del rendimiento en el aprendizaje

Obtener detalles sobre el avance y los resultados en el aprendizaje de un determinado contenido es algo influyente para la mayoría de los estudiantes. Esto se concluye de los datos obtenidos del gráfico de la *Ilustración 3-7* donde podemos apreciar que el 61% de los encuestados piensa que obtener una retroalimentación constante sobre el rendimiento del curso afecta positivamente.

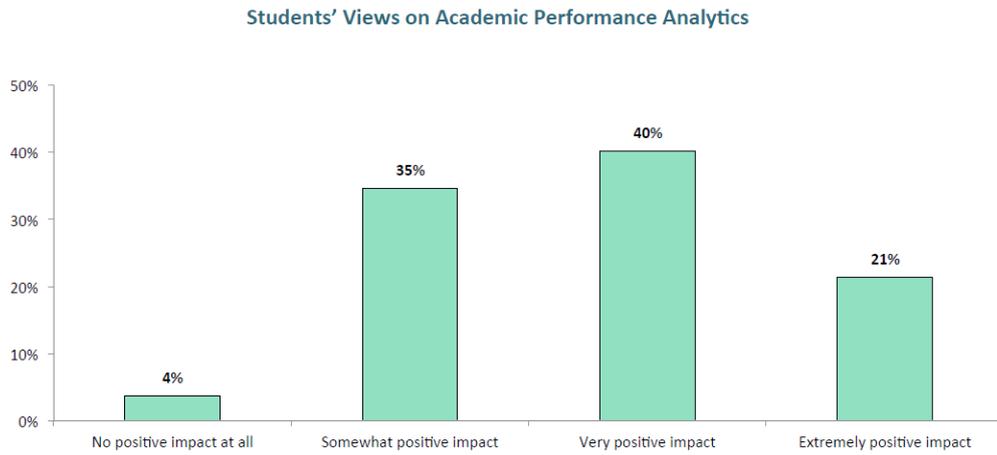


Ilustración 3-7: Influencia de la información de rendimiento [25]

No todas las aplicaciones que analizaremos en el apartado 3.4 incluyen estadísticas sobre el avance. Bajo mi punto de vista como usuario de este tipo de aplicaciones considero que disponer de estadísticas y información sobre el avance en el aprendizaje es algo motivador a continuar utilizando la aplicación.

3.4 Aplicaciones referentes de repaso espaciado

3.4.1 Duolingo

Es una aplicación enfocada únicamente en el aprendizaje de idiomas, incluye un sitio web y aplicaciones móviles para *Android* e *ios*. Esta aplicación tiene más de 100.000.000¹⁵ de instalaciones en la tienda de aplicaciones para dispositivos Android, Google Play¹⁶, lo que la sitúa como aplicación número 88 en el ranking de más descargadas a nivel mundial¹⁷.

Ventajas:

- ✓ Versión gratuita.
- ✓ 81 cursos de idiomas disponibles.
- ✓ Es muy práctica.
- ✓ Intuitiva y bien diseñada.
- ✓ Contenido de calidad.

Desventajas:

- X Únicamente para el aprendizaje de idiomas.
- X No dispone de explicaciones.
- X Pocas estadísticas sobre el avance.
- X Demasiado repetitivo.
- X No explica los errores cometidos.
- X No hay información sobre el método de repetición empleado y su efectividad
- X Contenido estático, no permite añadir contenido.

Duolingo	
Logotipo	
Año de lanzamiento	2011
Estado Actual	Activo
Tipo Comercial	Freemium ¹⁸
Plataformas	 Android Ios Windows Web

Tabla 3-1: Características Duolingo

¹⁵ Fuente: Google Play información pública de la aplicación. Fecha de consulta 07/07/2019.

¹⁶ **Google Play Store:** es una plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android.

¹⁷ Fuente: Google Play información pública de la aplicación. Fecha de consulta 15/07/2019.

¹⁸ **Freemium:** estrategia de fijación de precios mediante la cual un producto o servicio se proporciona de forma gratuita, pero se cobra dinero por características adicionales.

3.4.2 Menrise

Es una aplicación especializada en aprendizaje de idiomas, pero también dispone de contenido de otras temáticas. La aplicación está disponible para *Android, ios* y navegadores web. Esta aplicación tiene más de 10.000.000¹⁹ de instalaciones en la tienda de aplicaciones para dispositivos Android, Google Play

Ventajas:

- ✓ Versión gratuita.
- ✓ Gran número de cursos.
- ✓ Idiomas y otras temáticas de aprendizaje.
- ✓ Intuitiva y bien diseñada.
- ✓ Permite añadir contenido.

Desventajas:

- ✗ Hay contenido incorrecto.
- ✗ Puede ser muy repetitiva.
- ✗ Pocas estadísticas sobre el avance.
- ✗ No hay información sobre el método de repetición empleado y su efectividad.

3.4.3 Anki

Es una aplicación que permite estudiar cualquier cosa mediante la creación de tarjetas de aprendizaje. La aplicación está disponible para casi todas las plataformas del mercado. Esta aplicación tiene más de 1.000.000²⁰ de instalaciones en la tienda de aplicaciones para dispositivos *Android, Google Play*. Utiliza la versión SM2 del algoritmo *SuperMemo* con algunas variaciones de mejora.

Ventajas:

- ✓ Amplio número de plataformas.
- ✓ Se conoce el algoritmo de repetición empleado y su efectividad.
- ✓ Es gratuita.
- ✓ Permite añadir contenido.

Desventajas:

- ✗ Hay contenido incorrecto.
- ✗ Diseño poco atractivo.
- ✗ El uso no es intuitivo.

Menrise		
Logotipo		
		
Año de lanzamiento		
2010		
Estado Actual		
Activo		
Tipo Comercial		
Freemium		
Plataformas		
		
Android	Ios	Web

Tabla 3-2: Características Memrise

Anki		
Logotipo		
		
Año de lanzamiento		
2010		
Estado Actual		
Activo		
Tipo Comercial		
No comercial.		
Plataformas		
		
		
Android	Ios	Web Linux Windows

Tabla 3-3: Características Anki

¹⁹ Fuente: Google Play información pública de la aplicación. Fecha de consulta 15/07/2019.

²⁰ Fuente: Google Play información pública de la aplicación. Fecha de consulta 15/07/2019.

3.4.4 Lingvist

Es una aplicación únicamente especializada en aprendizaje de idiomas. La aplicación está disponible para *Android*, *Ios* y navegadores web. Esta aplicación tiene más de 1.000.000²¹ instalaciones en la tienda de aplicaciones para dispositivos *Android*, *Google Play*. *Lingvist* ha sido el ganador del premio *EdTechXEurope*²² en 2017.

Ventajas:

- ✓ Buen diseño
- ✓ Aplicación Intuitiva.
- ✓ Estadísticas sobre el rendimiento en el curso.

Desventajas:

- X Solo una modalidad de tarjetas de aprendizaje.
- X Muy limitada la versión gratuita.
- X No permite añadir contenido.
- X El algoritmo de repetición se desconoce y su efectividad.

Lingvist	
Logotipo	
Año de lanzamiento	2014
Estado Actual	Activo
Tipo Comercial	Si, Freemium
Plataformas	   Android Ios Web

Tabla 3-4: Características Lingvist

3.4.5 SuperMemo

Es la aplicación pionera en el género, desarrollada por el creador del algoritmo *SuperMemo*. Esta aplicación disponible para la gran mayoría de plataformas utiliza versiones comerciales del algoritmo *SuperMemo* que no han sido liberadas, en la actualidad la versión 17.

Ventajas:

- ✓ Amplio número de plataformas.
- ✓ Se conoce el algoritmo de repetición empleado y su efectividad.
- ✓ Permite añadir tarjetas de aprendizaje.

Desventajas:

- X Los cursos de calidad son de pago.
- X No es posible seleccionar la versión del algoritmo de repetición.
- X No muestra información clara sobre el aprendizaje.

SuperMemo	
Logotipo	
Año de lanzamiento	1987
Estado Actual	Activo
Tipo Comercial	Comercial
Plataformas	    Android Ios Web Windows

Tabla 3-5: Características SuperMemo

²¹ Fuente: Google Play información pública de la aplicación. Fecha de consulta 15/07/2018.

²² **EdTechXEurope**: es la cumbre líder que reúne a inversores de nivel ejecutivo, innovadores y personas influyentes de la industria de empresas educativas europeas e internacionales.

3.5 Comparativas aplicaciones referentes.

Para que de una forma visual y clara se pueda comparar las características de las aplicaciones de referencia analizadas, incluimos la

Tabla 3-6 donde comparamos las características que consideramos más importantes.

					
Versión gratuita	✓	✓	✓	✓	✓
Versión de pago	✓	✓	✗	✓	✓
Contenido gratuito de calidad	✓	✓	✗	✓ ^{*2}	✗
Limitación de uso diario gratuito	✗	✗	✗	✓	✗
Versión para Windows	✓	✗	✓	✗	✓
Versión para macOS	✗	✗	✗	✗	✗
Versión para Linux	✗	✗	✓	✓	✗
App móvil Android		✓	✓	✓	✓
App móvil iOS	✓	✓	✓	✓	✓
Versión web	✓	✓	✗	✓	✓
Permite crear contenido	✗ ^{*1}	✓	✓	✗ ^{*1}	✓
Información sobre su efectividad	✗	✗	✗	✗	✓
Usabilidad de la App móvil	Muy Buena	Buena	Baja	Buena	Media
Facilidad de uso	Muy Buena	Buena	Baja	Media	Media
Información sobre el avance del aprendizaje	Baja	Baja	Baja	Media	Baja

Tabla 3-6: Comparativa aplicaciones referentes

*1 No es posible crear contenido en la versión gratuita o la creación se encuentra en fase beta.

*2 El número de tarjetas que es posible responder de forma continuada está limitado diariamente en la versión gratuita de la aplicación.

3.6 Crítica al estado del arte, conclusiones y propuesta

Llegados a este punto, pretendemos lograr a una conclusión clara sobre los datos obtenidos en los apartados 3.3, 3.4 y 3.5 con el objetivo de justificar cuales son los principales aspectos que nuestra aplicación debe cumplir para ser de utilidad. Detallamos estos en los siguientes puntos:

- Fomentar el contenido multimedia, como audio y video, frente a largos textos.
- Mostrar estadísticas sobre los resultados de avance en el curso.

- Cubrir el mayor número de plataformas posibles.
- Emplear un algoritmo de repetición que aporte indicios sobre su efectividad.
- Permitir a los usuarios añadir contenido de forma sencilla.
- Obtener las mejores condiciones de diseño y usabilidad posibles.
- Diseñar una correcta adaptación de la visualización de los recursos para dispositivos de pantalla reducida.



Capítulo

4

TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS

4.1 Introducción

Durante la última década en la que he estado inmerso en el mundo tecnológico han nacido y quedado en el olvido multitud de tecnologías y herramientas para la construcción de aplicaciones. Su elección es un proceso complicado, y al que en ocasiones no dedicamos el tiempo necesario, aun sabiendo que seleccionar una tecnología errónea para el proyecto puede suponer su fracaso a corto o largo plazo. Para la realización de este proyecto se han integrado tecnologías y herramientas que permiten generar, a partir de una **base común de código**, aplicaciones para diferentes plataformas y dispositivos. Esto es posible gracias a que en los últimos años nuestras aplicaciones web son mucho más dinámicas y poderosas gracias a *JavaScript*, hemos movido mucho código que normalmente estaba en el servidor hacia nuestros navegadores, en ocasiones dejándonos con miles de líneas de código JavaScript conectándose con varios archivos *HTML* y *CSS*. Por ello cada vez más desarrolladores utilizamos diferentes *Framework JavaScript* como los que describiremos en este capítulo y que son utilizados para estructurar y conectar los diferentes archivos *HTML*, *CSS* y *JavaScript*.

Además, realizaré un repaso por las tecnologías y herramientas utilizadas para la construcción, analizando en profundidad las que considero más influyentes e importantes en el proyecto y comparando en algunos casos las tecnologías y herramientas empleadas con otras de su género, debatiendo el por qué su elección frente al resto de alternativas.

4.2 Aplicaciones Híbridas

En la actualidad existen principalmente dos plataformas móviles. Por un lado, *Android* sobre la cual se desarrollan aplicaciones principalmente en el lenguaje de programación *Java* y por otro lado *iOs* cuyas aplicaciones son desarrolladas en los lenguajes de programación *Objective-C* y *Swift*. Además de los lenguajes específicos de cada plataforma, desarrollar una aplicación implica conocer no solo el lenguaje sino también el *SDK* o kit de desarrollo de software específico de cada plataforma. Esto conlleva que el desarrollo de aplicaciones para plataformas móviles requiera de un elevado conocimiento específico para la plataforma sobre la que se esté desarrollando.

Este inconveniente puede ser superado mediante el desarrollo de aplicaciones móviles híbridas, una solución basada en *HTML JavaScript* y *CSS*. Estas aplicaciones son creadas una sola vez, pero nos permiten ser utilizadas tanto en *Android* como en *iOS* compartiendo la misma base de código. Esto es posible ya que las aplicaciones se ejecutan en un *WebView*, siendo este un componente común tanto en aplicaciones *iOS* como *Android*. Es este *WebView* el que se ejecuta dentro de un contenedor nativo o "carcasa nativa" para cada una de las plataformas. Además, mediante el uso de *plugin* o gracias a la compatibilidad que en los últimos años están alcanzando los navegadores web o *WebView* para acceder a los recursos del dispositivo desde el que son utilizados, tendremos acceso a recursos como la cámara o la ubicación del dispositivo entre otros. Podemos ver una representación de esto en la *Ilustración 4-1*

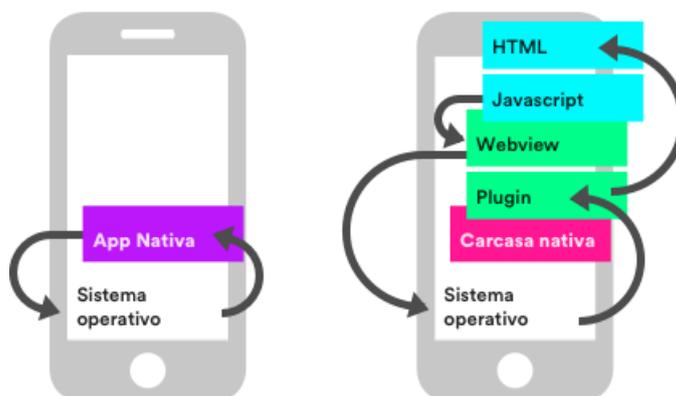


Ilustración 4-1: Arquitectura app híbrida vs nativa
<< Fuente: cuatroochenta.com>>

4.3 Aplicaciones Web Progresivas - PWA

Históricamente, las aplicaciones nativas (instaladas en el teléfono móvil) han sido capaces de ofrecer una experiencia de usuario mucho mejor que las webs, aun siendo estas diseñadas específicamente para dispositivos móviles [26]. El rendimiento o trabajar sin conexión nos hacían decantarnos por una

implementación nativa frente a una implementación web. En 2015, el ingeniero de *Google Chrome*²³, *Alex Russell*, define las aplicaciones web progresivas (desde ahora, PWA), como webs que aprovechan las nuevas características compatibles con navegadores modernos y que dotan a las web de un conjunto de nuevas características, haciendo que estas cumplan muchas de las expectativas que tenemos de las aplicaciones nativas [27]. Entre las expectativas que libran las PWA se encuentra:

- Disponibilidad independientemente de la conexión.
- Tiempos de carga más rápidos.
- Notificaciones Push (notificaciones, aun estando la aplicación cerrada).
- Atajo de pantalla de inicio (posibilidad de anclarlas como aplicaciones nativas al menú del dispositivo).
- Aspecto nativo (definir la orientación, visualizar a pantalla completa, etc...)

Pero debemos ver las PWA como más que un conjunto de excelentes características nuevas; son una forma de crear mejores sitios con el conjunto de las **mejores prácticas del desarrollo web** [26]. Además, al adoptar aplicaciones web progresivas como una alternativa frente a un desarrollo nativo, se pueden reducir los costos técnicos y operativos asociados con la construcción y soporte de múltiples aplicaciones para diferentes tipos de dispositivos [27].

Para comprender como hacen posible la adopción de las características descritas, es necesario comprender que es el *Service Worker* ya que este podemos denominarlo el corazón de una aplicación PWA.

Service Worker.

Podemos ver este como un script que se puede registrar para controlar las páginas del sitio. Una vez instalado, este trabaja fuera de cualquier ventana o pestaña del navegador incluso sin existir conexión entre el servidor web y el dispositivo [27]. Para comprender con mayor claridad que aporta el *Service Worker* podemos ver la *Ilustración 4-2* donde existe una capa entre la página y la web que puede responder a solicitudes. En la *Ilustración 4-3*, podemos apreciar como la capa indicada continua disponible aun sin conexión, esto posibilita que se continúen gestionando las diferentes pestañas y páginas del sitio devolviendo el contenido en caché.

²³ **Google Chrome:** es un navegador web de software privativo o código cerrado desarrollado por Google, aunque derivado de proyectos de código abierto.

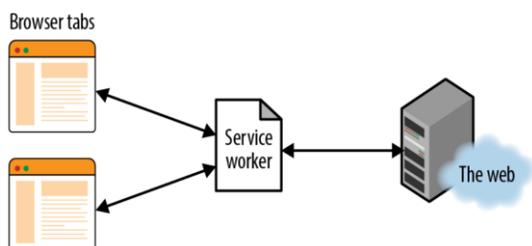


Ilustración 4-2 La web y el Service Worker [27]

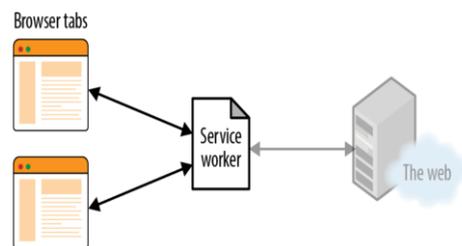


Ilustración 4-3 Páginas que se comunican con un Service Worker mientras el usuario está offline [27]

Esta capa ofrecida por el *Service Worker* además posibilita que un usuario que cerró todas las pestañas de la aplicación en el navegador, aún mantenga comunicación con el servidor como se representa en la *Ilustración 4-4*.

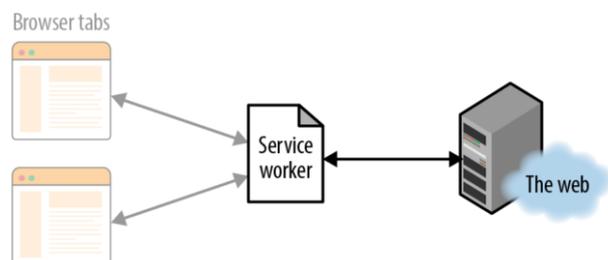


Ilustración 4-4 Service Worker que se comunica con un servidor después de que un usuario ha salido de la página [27]

Por último, hay que indicar que para permitir la ejecución de un *Service Worker* en un sitio web, debe ser servido a través de HTTPS²⁴, esto es un requisito indispensable por seguridad ya que, si alguien fuera capaz de registrar un *Service Worker* malicioso en una página web, podría secuestrar conexiones y redirigirlas a otro punto [26].

²⁴ HTTPS: es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hipertexto.

4.4 Framework empleados en la construcción del front-end

4.4.1 Apache Cordova

Podemos definir *Apache Cordova* como un entorno de desarrollo de aplicaciones para dispositivos móviles. Mediante este framework es posible construir aplicaciones para dispositivos móviles utilizando lenguajes como *CCS3*, *HTML5* y *JavaScript* en lugar de hacer uso de las APIs específicas de las plataformas *Android*, *iOS*, o *Windows Phone* que requerirían utilizar lenguajes específicos para cada una de las plataformas. Es decir, este *framework* permite mediante un código común, escrito en los tres lenguajes mencionados, generar versiones para las principales plataformas de dispositivos móviles. Esta peculiaridad hace que las aplicaciones resultantes sean conocidas como **aplicaciones híbridas**

Aunque las aplicaciones resultantes de este *framework* pueden tener una representación gráfica similar a las generadas de forma nativa para una plataforma en concreto, en realidad la representación gráfica se realiza mediante vistas web, en lugar de emplear las del propio *framework* de la plataforma en la que se esté ejecutando la aplicación. Pero no únicamente debemos ver este tipo de aplicaciones como web embebidas en un dispositivo móvil, ya que, como aplicaciones para su distribución, estas tienen acceso a las APIs nativas del dispositivo mediante una gran variedad de *plugin*²⁵, permitiéndonos de esta forma acceder por ejemplo a la cámara del dispositivo o al micrófono.

Apache Cordova	
Logotipo	
Desarrollador(es)	Apache Foundation Software
Género	Framework
Licencia	Apache 2.0
Estado Actual	Activo

Tabla 4-1: Características Cordova

²⁵ **Plugin**: aplicación (o programa informático) que se relaciona con otra para agregarle una función nueva y generalmente muy específica.

4.4.2 Electron Framework

Es un *framework* para la construcción de aplicaciones gráficas de escritorio, este permite generar aplicaciones utilizando lenguajes y tecnologías que normalmente son empleadas para generar el *front-end* y el *back-ends*²⁶ de aplicaciones web. Como en su propia web indican “Crea aplicaciones multiplataforma de escritorio con *JavaScript*, *HTML* y *CSS*”.

Actualmente es compatible con *Mac*, *Windows* y *Linux*, es decir, las aplicaciones de *Electron* pueden ser compiladas y correr en las tres plataformas principales.

En la actualidad aplicaciones distinguidas, como *Visual Studio Code*, desarrollado por *Microsoft* o *PhoneGap Desktop* producida por *Adobe*, han sido generadas mediante la utilización del motor gráfico de este *framework*.

Electron Framework
Logotipo

Desarrollador(es) GitHub
Género <i>Framework</i>
Licencia <i>MIT</i>
Estado Actual Activo

Tabla 4-2: Características *Electron*

4.4.3 Quasar Framework

Quasar es un *framework* desarrollado mediante *Vue*, permite a los desarrolladores escribir un único código que permite, generar aplicaciones que puedan ser compiladas generando como resultado sitios web PWA (*Progressive Web Apps*), aplicaciones móviles (*Android*, *iOS*, ...) a través de *Apache Cordova* o aplicaciones de escritorio multiplataforma mediante *Electron*.

Entre las ventajas destacadas de este *framework*, es que realiza una transpilación automática del código *JavaScript ECMAScript6*²⁷ lo que nos permite utilizar las últimas novedades del lenguaje, sin preocuparnos por la compatibilidad que aún no es total en todos los navegadores del mercado. Además de estas, este *framework* aporta muchas más técnicas y herramientas del desarrollo de vanguardia.

Quasar Framework
Logotipo
 Quasar
Desarrollador(es) quasar-framework.org
Género <i>Framework</i>
Licencia <i>MIT</i>
Estado Actual Activo

Tabla 4-3: Características *Quasar*

²⁶ **Font-end y Back-End:** en ingeniería de software, este término se refiere a la separación de preocupaciones entre la capa de presentación (*front-end*) y la capa de acceso a datos (*back-end*) de una pieza de software.

²⁷ **ECMAScript6:** es una especificación de lenguaje y actualmente un estándar. El lenguaje *JavaScript* es un dialecto de este estándar. Los motores de los navegadores tardan en implementar las nuevas especificaciones de ES por lo que se hace necesaria para usar funcionalidades de las últimas versiones de ES una transpilación del código.

4.5 Lenguajes de programación empleados

4.5.1 Introducción

Para la elección de los lenguajes de programación, uno de los criterios tenidos en cuenta, fue la curva de aprendizaje²⁸, ya que, en caso de seleccionar un lenguaje desconocido en el que el coste de aprendizaje fuese elevado, podría suponer un retraso en el ajustado calendario de entrega del proyecto. Por este motivo, se han empleado lenguajes de marcas conocidos como *HTML5* y *CSS3* en los que dispongo de cierta soltura, estos lenguajes principalmente son empleados para las vistas de la aplicación.

En la *Ilustración 4-5* podemos apreciar una gráfica obtenida del repositorio del proyecto en el que nos muestra los diferentes lenguajes de programación empleados junto con su proporción de uso.

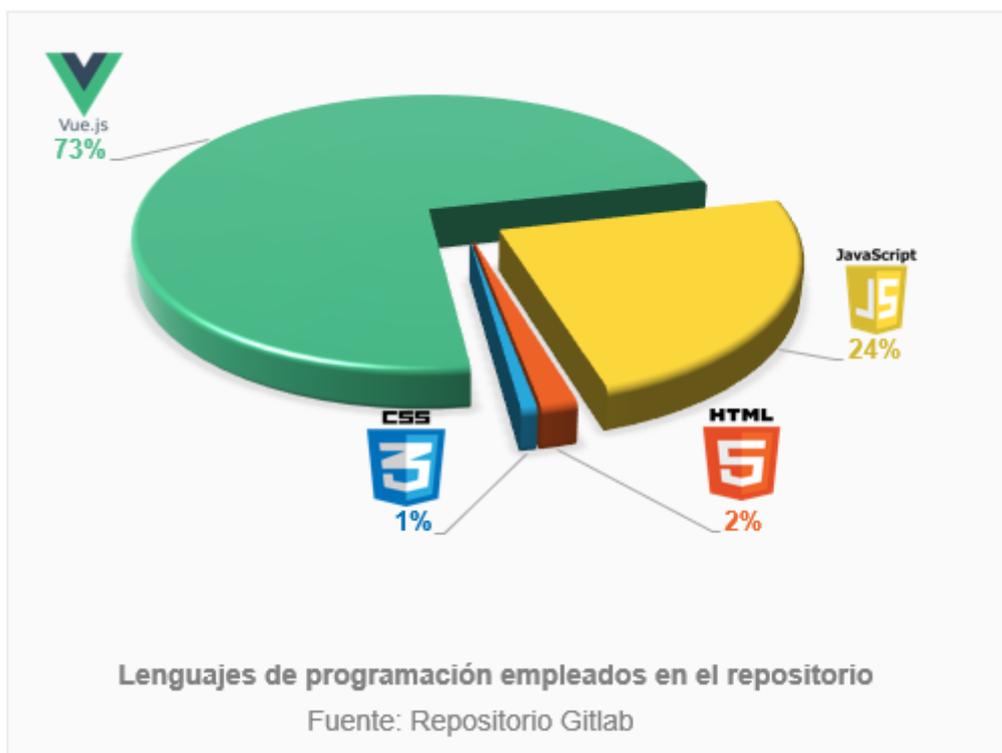


Ilustración 4-5: Lenguajes de programación utilizados en el proyecto

Además de *HTML* y *CSS*, se ha utilizado *JavaScript* como lenguaje de programación y *Vue* que como podemos apreciar en el anterior gráfico, representa el 73% del código del proyecto y que al inicio de este era un lenguaje totalmente desconocido para mí.

²⁸ **Curva de Aprendizaje:** una curva de aprendizaje describe el grado de éxito obtenido durante el aprendizaje en el transcurso del tiempo. Mientras más empinada sea la curva, mayor es la eficiencia del aprendizaje.

Por último, hay que destacar referente a *Vue* que, aunque en este proyecto y herramientas como *GitLab* lo distingamos como un lenguaje de programación diferenciado por su importancia, realmente se trata de una biblioteca de *JavaScript* como en la sección 4.5.2 de este capítulo veremos, y al que dedicaremos un mayor contenido ya que, como el anterior gráfico refleja este lenguaje supone un 73% del código total de nuestro proyecto frente al resto de lenguajes que conforman el mismo.

4.5.2 Vue

4.5.2.1 ¿Qué es Vue?

Vue es un *framework* de *JavaScript*, lanzado en 2014 por *Evan You*, un desarrollador que trabajaba en *Google* usando la conocida biblioteca *AngularJS*²⁹, como el mismo afirma en su blog “*Me imaginé, ¿y si pudiera extraer la parte que realmente me gustaba de Angular y construir algo realmente liviano?*” [28]. Es decir, *Vue* se centra en hacer que el desarrollo web sea menos obstinado y más fácil de entender para los desarrolladores. Su desarrollo es posible gracias a los numerosos patrocinadores que apuestan por este proyecto de software libre.

Vue
Logotipo

Desarrollador(es)
Evan You (尤雨溪)
Género
<i>Framework</i>
Licencia
<i>MIT</i>
Estado Actual
Activo

Tabla 1: Características de Vue

Este *framework* está construido desde cero, siendo posible su utilización de manera progresiva, lo que significa que, si solo necesitamos añadir una experiencia más intuitiva en una sección de una web ya escrita con otras bibliotecas, es posible utilizar este *framework* de una manera sencilla e integrarlo solo para la funcionalidad requerida, o si por el contrario se requiere comenzar un proyecto desde un principio *Vue* nos aporta el ecosistema necesario para construir una aplicación de forma estructurada y sencilla.

Otra de sus ventajas es lo liviano frente a otros *frameworks* de su género. En su web oficial existe una comparativa frente *Angular* donde se afirma que las versiones actuales de *Angular* se encuentran entorno a (~65KB) en un proyecto inicial frente a los (~30KB) de un proyecto *Vue* [29]. Esto es algo que debemos tener en cuenta ya que debemos recordar que este *framework*

²⁹ **AngularJS**: es un *framework* para la construcción de aplicaciones web de una sola página. Su objetivo es aumentar y facilitar las posibilidades de las aplicaciones en el navegador.

será empleado en navegadores o en dispositivos móviles en los que la disponibilidad de memoria está limitada.

4.5.2.2 ¿Por qué Vue?

Al comienzo de este proyecto se analizaron diferentes *framework* de aplicaciones móviles, uno de los más conocidos en la actualidad es *Ionic*³⁰, este *framework* está construido sobre *Angular*³¹, lo que hacía necesario en caso de su utilización el aprendizaje de *Angular*. Como anteriormente indicamos en el índice de este capítulo uno de los principales criterios tenidos en cuenta para la elección de los diferentes lenguajes y tecnologías empleadas en la construcción del proyecto, fue la curva de aprendizaje, *Angular* al igual que otros *framework* como *React*³² tienen una curva de aprendizaje elevada, como así se afirma en la documentación oficial de *Vue* [29]. Por lo que decidí buscar otras alternativas. En dicha búsqueda encontré *Vue*, un *framework JavaScript* que se está volviendo popular muy rápidamente, llegué a esta conclusión tras analizar las tendencias de *Google Trends*³³. Vemos esto en la gráfica de la *Ilustración 4-6* donde se denota el interés sobre *Vue* el cual, aumento en los últimos dos años entorno al 50%.

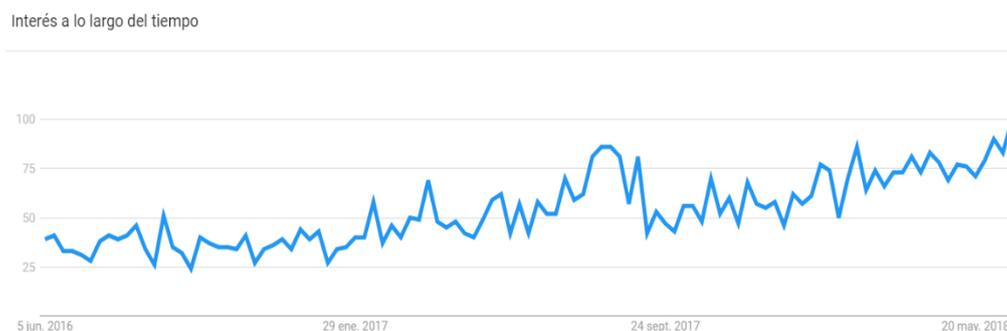


Ilustración 4-6: Interés a lo largo del tiempo Vue.js
<< Fuente: Google Trends >>

La gráfica anterior obtenida de *Google Trends* nos da una visión clara sobre el interés de *Vue*, pero no es el criterio más adecuado de tomar una decisión sobre la tecnología que emplear en el proyecto, por este motivo continúe indagando, analizando el número de descargas realizadas desde el repositorio *Npm*³⁴ durante los últimos dos años, *Vue*, actualmente está superando el número de descargas del conocido *framework AngularJS* desarrollado por *Google* como se puede apreciar en el siguiente gráfico de la *Ilustración 4-7*.

³⁰ **Ionic Framework:** es un SDK completo de código abierto para el desarrollo de aplicaciones móviles híbridas, en la actualidad es uno de los *framework* más populares en su categoría.

³¹ **Angular:** es un *framework* para la construcción de aplicaciones web de una sola página. Angular es la evolución de *AngularJS*, aunque incompatible con su predecesor.

³² **React:** es una biblioteca JavaScript pensada para ayudar en el proceso de creación de aplicaciones Web de una sola página.

³³ **Google Trends:** es una herramienta de Google que muestra los términos de búsqueda más populares.

³⁴ **Npm:** es el manejador de paquetes por defecto para Node.js, Mediante npm es posible manejar las dependencias para una aplicación.

Downloads in past 2 Years ▾

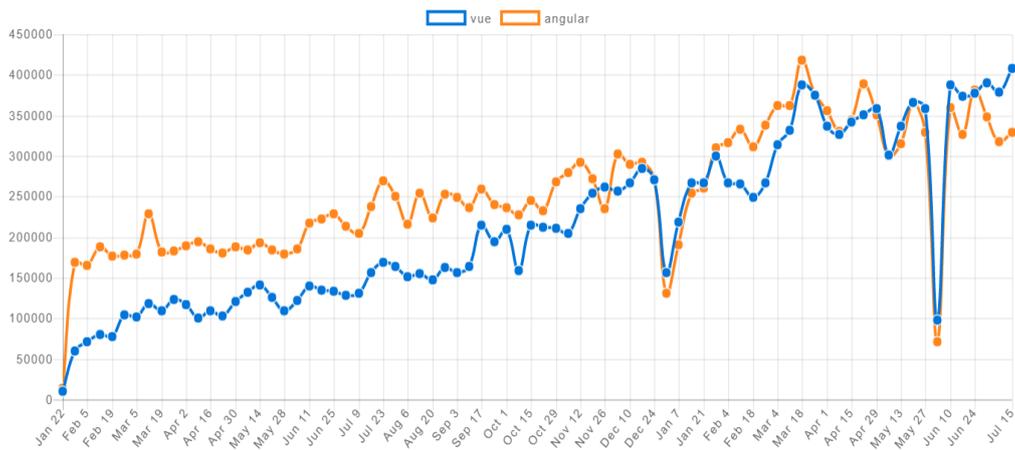


Ilustración 4-7: Descargas Vue.js vs AngularJS
<<Fuente:www.npmtrends.com>>

Tras confirmar que el interés de este *framework* va en aumento, y que podía ser una buena opción, decidí analizar cuestiones como el rendimiento frente a los dos referentes en la actualidad *Angular* y *React*, analicé los resultados del estudio de rendimiento realizado por el blog de *Stefan Krause* que, durante los últimos años ha realizado comparativas sobre los diferentes *framework JavaScript* [30]. Estos resultados que podemos ver en la *Ilustración 4-8*.

Nombre	angular-v5.0.0-	vue-v2.5.3-	react-v16.1.0-redux-
crear filas Duración para crear 1000 filas después de la página cargada.	185,7 ± 7.8 (1,4)	169,2 ± 3,6 (1,2)	206,2 ± 9,0 (1,5)
reemplazar todas las filas Duración para actualizar las 1000 filas de la tabla (con 5 iteraciones de calentamiento).	179,3 ± 6,5 (1,2)	161,8 ± 3,9 (1,1)	175,1 ± 4,2 (1,2)
actualización parcial Es hora de actualizar el texto de cada décima fila (con 5 iteraciones de calentamiento) para una tabla con 10k filas.	73,5 ± 4,9 (1,0)	168,1 ± 7,4 (2,3)	97,8 ± 5,6 (1,4)
Seleccionar fila Duración para resaltar una fila en respuesta a un clic en la fila. (con 5 iteraciones de calentamiento).	7,6 ± 4,0 (1,0)	9,8 ± 2,5 (1,0)	10,1 ± 3,5 (1,0)
filas de intercambio Tiempo para intercambiar 2 filas en una tabla 1K. (con 5 iteraciones de calentamiento).	20,1 ± 4,2 (1,3)	21,8 ± 4,5 (1,4)	19,0 ± 4,4 (1,2)
eliminar fila Duración para eliminar una fila (con 5 iteraciones de calentamiento).	46,1 ± 2,6 (1,1)	52,5 ± 1,8 (1,2)	49,5 ± 1,8 (1,1)
crear muchas filas Duración para crear 10,000 filas	1682,0 ± 53,1 (1,3)	1521,4 ± 55,7 (1,1)	2048,5 ± 58,8 (1,5)

Nombre	angular-v5.0.0-	vue-v2.5.3-	react-v16.1.0-redux-
memoria lista Uso de la memoria después de la carga de la página.	6,7 ± 0,1 (2,3)	3,6 ± 0,1 (1,2)	4,2 ± 0,1 (1,4)
ejecutar memoria Uso de memoria después de agregar 1000 filas.	10,5 ± 0,0 (2,9)	7,2 ± 0,0 (2,0)	8,5 ± 0,0 (2,3)
actualización eatch 10ma fila para 1k filas (5 ciclos) Uso de memoria después de hacer clic en actualizar cada décima fila 5 veces	10,6 ± 0,0 (2,9)	7,3 ± 0,0 (2,0)	9,6 ± 0,0 (2,6)
Reemplace 1k filas (5 ciclos) Uso de memoria después de hacer clic en crear 1000 filas 5 veces	10,8 ± 0,0 (3,1)	7,3 ± 0,0 (2,0)	11,0 ± 0,0 (3,1)
creando / borrando 1k filas (5 ciclos) Uso de memoria después de crear y borrar 1000 filas 5 veces	7,1 ± 0,0 (2,3)	3,8 ± 0,0 (1,2)	5,8 ± 0,0 (1,9)
agregar filas a la tabla grande Duración para agregar 1000 filas en una tabla de 10,000 filas.	257,6 ± 11,1 (1,2)	338,4 ± 10,3 (1,6)	300,7 ± 30,9 (1,4)
filas claras Duración para borrar la tabla llena con 10.000 filas.	360,3 ± 16,4 (2,0)	240,9 ± 11,4 (1,4)	227,7 ± 8,6 (1,3)
tiempo de inicio Tiempo de carga, análisis y puesta en marcha	88,8 ± 2,9 (3,6)	48,4 ± 2,4 (2,0)	68,6 ± 2,3 (2,8)
ralentización geométrica media	1,38	1,38	1,39

Ilustración 4-8: Resultado comparativa rendimiento Vue .<<Fuente:stefankrause.net >>

Uno de los factores de la comparativa y que considero de vital importancia, y que podemos apreciar en penúltima fila de la *Ilustración 4-8* es el tiempo de inicio de *Vue* frente a *Angular* o *React* que incluso se reduce a la mitad. Este es un factor que considero clave en este proyecto, ya que *esté framework* será embebido en una aplicación para dispositivos móviles y por lo tanto el tiempo de carga al inicio de la aplicación debería ser lo más liviano posible.

Angular y *React*, son referentes en la construcción de aplicaciones y, ambos están respaldados por dos grandes empresas, *Google* en el caso de *Angular* y, *Facebook* en el caso de *React*. Por este motivo analicé que proyectos respaldan el uso de *Vue*, encontrando entre otros muchos, algunos tan conocidos como *Laravel*³⁵ o *GitLab*.

³⁵ **Laravel:** Es un framework con gran influencia en la actualidad para la construcción de aplicaciones y servicios web con PHP. Tiene como objetivo desarrollar código utilizando una sintaxis clara y una estructuración MVC.

Como conclusión he de indicar que el rápido crecimiento de *Vue*, que viene condicionado por su facilidad de uso, su baja curva de aprendizaje y su utilización en proyectos importantes hizo que optase por él, frente a otras opciones como *React*, *AngularJS* o *Angular*.

4.5.2.3 Librerías del ecosistema *Vue*

Vue es un *framework* progresivo, es decir, podemos utilizarlos para pequeños proyectos utilizando únicamente su biblioteca principal o utilizarlo en aplicaciones sofisticadas. Esto es debido al ecosistema de librerías que dan apoyo en ciertas funcionalidades o características que puedan ser demandadas.

En los siguientes apartados se recogen las bibliotecas más importantes de ecosistema *Vue* utilizadas en el proyecto.

4.5.2.3.1 Axios

En muchas ocasiones se requiere consumir datos de una web API³⁶ como sucede en nuestra aplicación. Para este cometido *Vue* dispone de una librería denominada *Axios* y que actúa como un cliente HTTP basado en promesas³⁷.

Esta librería permite realizar solicitudes contra una web API completamente configurables y recibir la respuesta de forma sencilla y fácil de procesar. Su uso está basado en promesas, lo que permite generar un código asíncrono de forma entendible y ordenado.

4.5.2.3.2 Vuex

Vuex es una librería para la gestión de estados en *Vue*. Su funcionamiento se basa en la arquitectura del patrón *singleton*³⁸, permitiéndonos tener datos centralizados y accesibles desde cualquier lugar de la aplicación y sin que estos datos puedan ser modificaciones sin control. *Vuex* dispone de mecanismos que garantizan que el estado solo puede mutar de manera controlada y predecible.

La comunidad *Vue* recomienda utilizar en aplicaciones de mediana a gran escala. Esto es debido a que el uso de esta librería aporta cierta complejidad por lo que su uso debe valorarse como una compensación entre la productividad a corto y largo plazo.

³⁶ **Web API:** En la web, las API's son publicadas por sitios para brindar la posibilidad de realizar alguna acción o acceder a alguna característica o contenido que el sitio provee.

³⁷ **Promesas:** en javascript una promesa es un objeto que representa la terminación o el fracaso de una operación asíncrona.

³⁸ **Patrón singleton:** en ingeniería del software es un patrón de diseño que permite restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

4.5.3 JavaScript

Conocido también por sus siglas *JS*, es un lenguaje de programación, interpretado³⁹ con funciones de primera clase⁴⁰ y orientado a objetos⁴¹. Junto con *HTML* y *CSS*, *JavaScript* es una de las tres tecnologías principales de la *World Wide Web*⁴² [31]. El uso más común de JavaScript es escribir funciones embebidas o incluidas en páginas *HTML* y que interactúan con el *Document Object Model* (DOM o Modelo de Objetos del Documento) de la página web. Aunque en los últimos años, han proliferado, entornos que utilizan *JavaScript* fuera del navegador como *Node.js*⁴³.

Hasta la aparición de *Ajax*⁴⁴, muchos programadores profesionales denigraban el lenguaje porque, entre otras razones, su público objetivo consistía en autores web y otros "aficionados" tal y como afirma *Douglas Crockford*, desarrollador de *JavaScript* [32]. Durante esta última década, han proliferado numerosos *framework* y bibliotecas que aportan a JavaScript un conjunto de buenas prácticas para su uso, dentro y fuera del navegador Web.

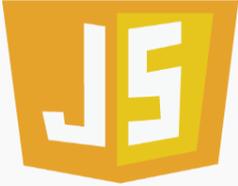
JavaScript
Logotipo JAVASCRIPT 
Desarrollador(es) Fundación Mozilla, Ecma International, Netscape
Ultima versión estable ECMAScript 2017
Año publicación 1995
Estado Actual Activo

Tabla 4-4: Características de JavaScript

³⁹ **Lenguaje Interpretado:** es el lenguaje cuyo código no necesita ser procesado mediante un compilador, eso significa que el ordenador es capaz de ejecutar la sucesión de instrucciones dadas por el programador sin necesidad de leer y traducir exhaustivamente todo el código.

⁴⁰ **Funciones de primera clase:** las funciones se tratan como cualquier otra variable, es decir, una función puede pasarse como argumento a otras funciones, puede ser devuelta por otra función o asignarse como un valor a una variable.

⁴¹ **Lenguaje Orientado a Objetos (OOP):** lenguajes que permiten contener datos, en forma de campos (atributos); y código, en forma de procedimientos (métodos).

⁴² **World Wide Web:** es básicamente un medio de comunicación de texto, gráficos y otros objetos multimedia a través de Internet, es decir, la web.

⁴³ **Node.js:** entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor, basado en el lenguaje de programación ECMAScript.

⁴⁴ **Ajax:** es una técnica de desarrollo web para crear aplicaciones interactivas que permiten comunicación asíncrona en segundo plano con el servidor.

4.5.4 HTML

De sus siglas en inglés *Hypertext Markup Language* (*HTML*). Es un lenguaje de marcado⁴⁵ para la elaboración de páginas web estandarizado por la *W3C*⁴⁶.

La última versión estable, denominada *HTML5* se publicó como una recomendación de la *W3C* en 2014, en esta versión se han añadido y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas webs compatibles con distintos navegadores y plataformas.

HTML		
Logotipo		
		
Desarrollador(es)		
World	Wide	Web
Consortium		
Última versión estable		
W3C HTML5.2		
Año publicación		
1993		
Estado Actual		
Activo		

Tabla 4-5: Características de HTML

⁴⁵ **Lenguaje de marcado:** es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación.

⁴⁶ **W3C:** es un consorcio internacional que genera recomendaciones y estándares que aseguran el crecimiento de la World Wide Web a largo plazo

4.5.5 CSS

Las hojas de estilo en cascada, de sus siglas en inglés *Cascading Style Sheets (CSS)*. Son un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado como *HTML*.

Junto con *HTML* y *JavaScript*, *CSS* es una tecnología usada por los sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y móviles. *CSS* tiene una sintaxis simple y usa varias palabras clave en inglés para especificar los nombres de varias propiedades de estilo. Una hoja de estilo consiste en una lista de reglas. Cada regla o conjunto de reglas consta de uno o más selectores y un bloque de declaración.

Durante los últimos años, han proliferado gran cantidad de *framework CSS* que ofrecen una serie de opciones preparadas para diseñar rápidamente páginas web, uno de estos *framework* es *Bootstrap*. Aunque su uso es común entre los desarrolladores, existen retractoros del uso de estas librerías de estilo ya que su uso implica tener muchas características que pueden no ser utilizadas en el estilo del sitio, lo que puede afectar al rendimiento [33].

CSS		
Logotipo		
		
Desarrollador(es)		
World	Wide	Web
Consortium		
Ultima versión estable		
W3C CSS3		
Año publicación		
1996		
Estado Actual		
Activo		

Tabla 4-6: Características de HTML

4.6 Firesabase como “back-end as a service”

4.6.1 Introducción

Las aplicaciones web y móviles requieren en su mayoría de características en su *backend* muy similares, características como la autenticación, el almacenamiento, la integración con redes sociales o las notificaciones push⁴⁷ entre otras son características comunes en las aplicaciones actuales. Es por este motivo en los últimos años han surgido empresas que proporcionan plataformas que brindan estos servicios, estas son conocidos como *Backend as a Service*, o por su siglas *BasS*.

Uno de los motivos que me hizo decidirme por *Firebase* como *backend* de la aplicación, es el ahorro en tiempo de desarrollo [34] ya que la mayoría de los servicios que necesito ya están implementados y solo necesito consumirlos. Para esto, la mayoría de los proveedores ofrecen un conjunto de herramientas conocidos como *SDK*⁴⁸ disponibles para ser integrados en el *frontend*⁴⁹ de nuestra aplicación y el cual permite hacer uso de forma sencilla de las funcionalidades y servicios que ofrece el *BasS*.

Por estos motivos y tras analizar varios proveedores de *BasS* como *AWS Amplify*⁵⁰ o *Parse Server*⁵¹, considero que *Firebase* aporta tras mi análisis el mayor catálogo de servicios que actualmente necesita o puede necesitar la aplicación. Además, es un servicio de *Google* encontrándose totalmente integrado y compartiendo funcionalidades con *Google Cloud*⁵² [35].

Aun así y habiendo apostado por un *Backend as a Service*, decir que como podremos ver en la sección 6.2 la arquitectura de la aplicación se definió en capas, separando la integración que existe con el *SDK* de *Firebase*, lo cual posibilitaría en un futuro, si fuese necesario cambiar a otro tipo de *backend* desarrollado específicamente para esta aplicación con el menor impacto posible.

⁴⁷ **Notificación push:** es una forma de comunicación a menudo en preferencias de información a medida. Es decir, un modelo publicador/suscriptor.

⁴⁸ **SDK:** de sus siglas en ingles Software Kit Developer, es un conjunto de herramientas para el desarrollo de aplicaciones o servicios.

⁴⁹ **Frontend:** Pueden traducirse al español como interfaz, frontal final, es decir en una separación de intereses entre una capa de presentación y una capa de acceso a datos, el frontend sería la parte del software que interactúa con los usuarios

⁵⁰ **AWS Amplify:** es la plataforma de desarrollo de Amazon web services para la creación de aplicaciones móviles. Ofrece servicios como la autenticación de los usuarios el almacenamiento de datos o el análisis de las métricas de la aplicación entre otros.

⁵¹ **Parse Server:** es una comunidad de código abierto que permite ofrecer el backend móvil como servicio. Proporciona características como la administración de usuarios o las notificaciones push entre otros. Es posible alojar esta plataforma de forma privada o utilizarla mediante diversos proveedores que la ofrecen.

⁵² **Google Cloud:** es una plataforma que ha reunido todas las aplicaciones de desarrollo web que Google estaba ofreciendo por separado.

4.6.2 Firebase

Podemos definir *Firebase* como una plataforma para el desarrollo de aplicaciones para dispositivos móviles y web. Ofrece en la modalidad de *Backend as a Service* diferentes servicios que pueden ser consumidos por los desarrolladores facilitando y agilizando la puesta en marcha de nuevos proyectos web o móvil.

Entre las ventajas de su utilización frente a otras alternativas, destaca la buena documentación que mantienen y la activamente colaboración de sus desarrolladores en plataformas como *Github*⁵³ y *StackOverflow*⁵⁴.

Firebase fue desarrollada en 2012 y adquirida por Google en 2014 y propietario de la plataforma en la actualidad.

Firebase	
Logotipo	
Desarrollador(es)	Google
Género	<i>Backend as a Service</i>
Licencia	<i>Propietario</i>
Estado Actual	Activo

Tabla 4-7: Características *Firebase*

En su plataforma se ofrecen entre otros los siguientes servicios [35]:

- **Auth:** proporciona una solución de identidad de extremo a extremo, es compatible con la autenticación con cuentas de correo electrónico y contraseña, autenticación de teléfono o cuentas de *Facebook* o *Google*. Aporta un panel de control desde el que es posible realizar una gestión de los usuarios dados de alta en la aplicación.
- **Cloud Firestore:** es una base de datos NoSQL, flexible y escalable, permite mantener los datos sincronizados entre apps cliente a través de agentes de escucha en tiempo real y ofrece asistencia sin conexión para dispositivos móviles y la Web. Hablaremos en detalle de esto en la *sección 4.7.2*.
- **Cloud Messaging:** es una solución de mensajería multiplataforma que te permite enviar mensajes de forma segura y gratuita.
- **Hosting:** es un servicio de hosting de contenido web con nivel de producción orientado a programadores. Permite mediante un comando de forma sencilla y rápida entregar en una CDN (red de distribución de contenidos).
- **Cloud Storage:** es un servicio de almacenamiento de objetos que permite almacenar y publicar contenido generado por los usuarios como fotos o videos.

⁵³ **Github:** GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para proyectos alojados en de forma pública, aunque utilizando una cuenta de pago, también permite hospedar repositorios privados.

⁵⁴ **StackOverflow:** es un sitio web utilizado por una comunidad de desarrolladores informáticos, en la cual otros desarrolladores pueden encontrar soluciones a problemas de programación en diferentes lenguajes.

- **Test Lab:** proporciona una infraestructura de prueba de apps basada en la nube. Permite probar la app para Android o iOS en una amplia variedad de dispositivos y configuraciones, y ver los resultados.
- **Analytics:** es una solución que proporciona capacidad para generar informes sobre el comportamiento de la aplicación y de los usuarios.

Por último, hay que destacar que para el desarrollo de esta aplicación se hacen uso de los servicios de Autenticación, *Cloud Firestore* y Hosting de entre los indicados.

4.7 Tecnologías utilizadas para la persistencia de datos

4.7.1 Antecedentes para la elección de la base de datos

La elección de la base de datos empleada para el proyecto fue un proceso delicado, ya que, en mi experiencia como desarrollador durante estos años, siempre utilicé bases de datos relacionales. Pero a diferencia del resto de proyectos en los que participé, este proyecto tenía dos aspectos que me hacían dudar en la elección de una base de datos tradicional. El primero de ellos, el rendimiento, y el segundo, es que algunos de los datos que necesitaba almacenar no cumplen una estructura fija.

En relación con el primero de los aspectos, el rendimiento, consideré que es algo que debía tener en cuenta. Optar por una base de datos relacional, implicaría que gran parte de las consultas de la aplicación serían focalizadas en una única tabla que almacenase todas las tarjetas de aprendizaje, esto a largo plazo podría ocasionar problemas de rendimiento, cuando el número de tarjetas almacenadas en esta tabla aumentase, lo que nos llevaría obligados a optar por un particionado de esta tabla para mejorar el rendimiento de las consultas y los procesos de carga. Por el contrario, en algunas bases de datos documentales *NoSQL*, **el rendimiento de las consultas, es proporcional al tamaño del conjunto de resultados, no al del conjunto de datos** [36].

La otra de las características que nos llevó a decantarnos por una base de datos *NoSQL* frente a una base de datos relacional, es que las *NoSQL* documentales, permiten que los documentos almacenados en una misma colección **no se ajusten a un esquema estándar**, es decir, no tienen por qué contener las mismas secciones y atributos. Por ejemplo, en nuestro modelo de datos, la colección *card* contiene las diferentes tarjetas de aprendizaje de una determinada unidad de un curso, estas tarjetas de aprendizaje se diferencian unas de otras, existiendo tarjetas tipo test, tarjetas de traducción, etc... En la *Tabla 4-8* vemos un ejemplo de dos documentos de la colección *card*, ambos documentos contienen una parte de estructura similar, representada en color azul y otra diferente, representada en color morado.

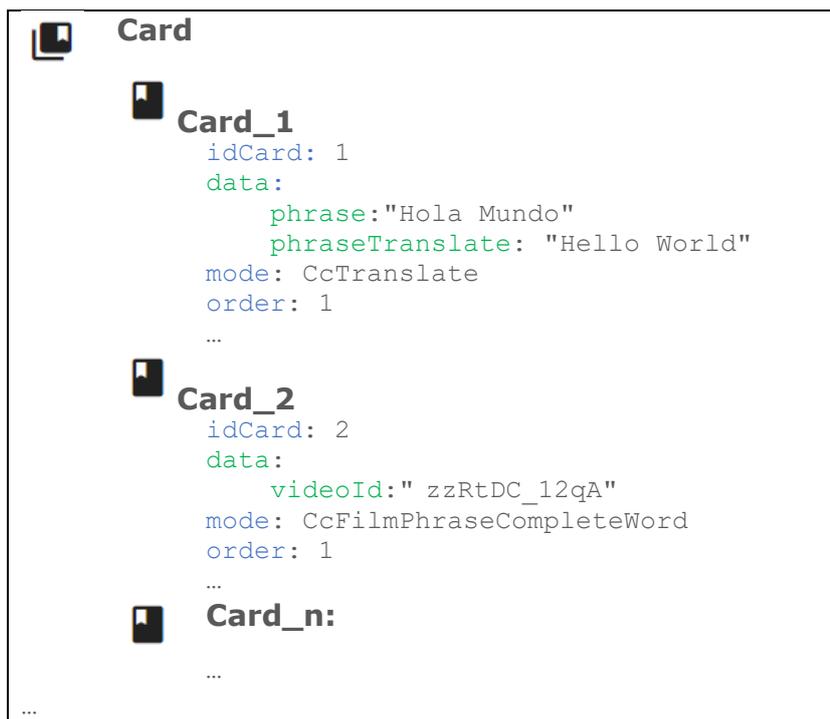


Tabla 4-8: Diferencias de esquemas de una misma colección

Estos dos aspectos destacados de las bases de datos NoSQL junto con el **interés por aprender una nueva modalidad de bases de datos** que en la actualidad se encuentra en auge, fue la que me llevo a decidirme por una base de datos como también es conocida "no sólo SQL".

4.7.2 Cloud Firestore, el motor y la base de datos

Cloud Firestore [37] es una base de datos *NoSQL* **orientada a documentos**. La diferencia de una base de datos orientada a documentos frente a una base de datos SQL es que, en lugar de almacenarse los datos en tablas y filas, los datos son almacenados mediante colecciones de documentos.

Como en la propia documentación del producto se indica, "*Cloud Firestore está optimizado para almacenar grandes colecciones de documentos pequeños*". Esto va en sintonía con nuestro modelo de datos ya que la principal característica de la aplicación y sobre la que más operaciones de escritura y lectura recaerán, es sobre la colección denominada *Card*, esta colección se corresponde con las tarjetas de aprendizaje empleadas en la aplicación y que, podemos definir como pequeñas unidades de conocimiento.

En la *Tabla 4-9* detallamos las características principales de *Cloud Firestore* y que consideramos de importancia para el proyecto.

Características Principales de Cloud Firestore	
Almacena datos en documentos organizados en colecciones	
✓	Permite almacenar subcolecciones dentro de documentos.
Compatibilidad sin conexión y en tiempo real.	
✓	SDK IOS.
✓	SDK Android
✓	SDK Clientes Web
Consultas	
✓	Consultar indexadas con ordenamiento y filtrado compuesto.
✓	Permite consultar subcolecciones en lugar de una colección entera.
✓	Las consultas se indexan de forma predeterminada: El rendimiento de las consultas es proporcional al tamaño del conjunto de resultados, no del conjunto de datos.
Confianza y rendimiento	
✓	Aloja los datos en varios centros de datos de distintas regiones, lo que garantiza una escalabilidad global y una confianza sólida
Seguridad	
✓	Se utilizan reglas de seguridad definidas en la propia base de datos que restringe consultas, eliminaciones y acciones.
✓	La validación de datos ocurre de forma automática.

Tabla 4-9: Características Cloud Firestore [36]

4.8 APIS utilizadas en el sistema

4.8.1 Pixabay API

Pixabay es un sitio web para el intercambio de fotos de alta calidad (fotos, imágenes, ilustraciones y gráficos vectoriales). Estas imágenes en este sitio son registradas en el dominio público según Licencias *Creative Commons*.

Este sitio Web dispone de una interfaz *RESTful*⁵⁵ para buscar y recuperar imágenes y videos gratuitos publicados en *Pixabay* devolviendo esta, objetos codificados en *JSON*⁵⁶.

4.8.2 YouTube Data API

Es una API de datos de la plataforma de video *YouTube*, permite cargar videos y administrar configuraciones de un canal. El uso de la API nos permite la búsqueda de videos que coincidan con los términos especificados por el usuario desde la propia aplicación, obteniendo de estos la información para que puedan ser embebidos como contenido.

⁵⁵ **RESTful**: en la actualidad se usa este término, para describir cualquier interfaz entre sistemas que utilice HTTP para obtener datos o indicar la ejecución de operaciones sobre los datos, en cualquier formato (XML, JSON, etc).

⁵⁶ **JSON**: JavaScript Object Notation. Es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript.

4.9 Herramientas utilizadas para el desarrollo

4.9.1 Git

Es un software de control de versiones, permite, rastrear cambios en archivos y coordinar el trabajo en esos archivos entre varias personas. Se utiliza principalmente para la gestión de código fuente en el desarrollo de software.

Este control de versiones se diferencia con otros controles de versiones, en que cada directorio de Git de cada ordenador es un repositorio completo con historial y capacidad de seguimiento de cambios, esto es independiente del acceso a la red o servidor central [38].

Por último, indicar que para la realización de este TFG se hace uso de la estrategia de ramificación conocida como *Gitflow* y que describe un conjunto de procedimientos que

los miembros de un equipo deben seguir para conseguir un proceso de desarrollo software administrado [39] [40] [41]. La estrategia de ramificación es explicada en detalle en el apartado 4.4 del CTFG que acompaña este trabajo.

Git	
Logotipo	
Desarrollador(es)	<i>Software Freedom Conservancy</i>
Licencia	<i>GNU GPL v2</i>
Año Lanzamiento	2005
Estado Actual	Activo

Tabla 4-10: Características Git

GitLab

GitLab es un servicio de alojamiento basado en web para los proyectos que utilizan el sistema de control de versiones **Git**. Además de gestor de repositorios, el servicio ofrece también alojamiento de *wikis* y de, entrega continua e implementación continua dentro de la misma interfaz web.

Actualmente existen dos versiones distintas, *GitLab CE (Community Edition)* y *GitLab EE (Enterprise Edition)*, esta última con funcionalidades adicionales. Para este proyecto, utilizaremos la capa gratuita que ofrece *GitLab.com*.

Este servicio de alojamiento es utilizado por importantes compañías como son *IBM, Sony, NASA* o *Intel*.

GitLab	
Logotipo	
Desarrollador(es)	GitLab Inc.
Licencia	MIT
Año Lanzamiento	2011
Estado Actual	Activo

Tabla 4-11: Características *GitLab*

4.9.2 Visual Studio Code

Es un Editor de código fuente desarrollado por Microsoft disponible para plataformas Windows, Linux y MacOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

Entre las ventajas que hace que muchos desarrolladores web nos decidamos por él, frente a otros editores, esta su rapidez, sencillez y potencia. Podemos decir tras su uso, que es un software liviano, pero a la vez potente, permitiendo añadir extensiones para realizar tareas que no estén incluidas en el paquete original.

VS Code	
Logotipo	
Desarrollador(es)	Microsoft
Licencia	MIT
Año Lanzamiento	2015
Estado Actual	Activo

Tabla 4-12: Características *VS Code*

4.9.3 Balsamiq Mockups

Balsamiq Mockups es una aplicación que permite diseñar maquetas, también denominadas Mockups, siendo estas, representaciones del diseño con poca o ninguna funcionalidad que permite hacer pruebas de diseño y evaluar el uso por parte de los usuarios.

Balsamiq Mockups
Logotipo  balsamiq®
Desarrollador(es) balsamiq.com
Licencia Propietario
Estado Actual Activo

Tabla 4-13: Características Balsamiq

4.9.4 Visual Paradigm Online

Es la versión online de la conocida herramienta de modelado UML *Visual Paradigm* (VP-UML). En su versión online permite realizar diagramas de negocios, técnicos y generales, incluyendo diagramas de flujo, UML o mapas mentales entre otros. A diferencia de otro tipo de herramientas que realizan la misma funcionalidad, esta versión online destaca por su simplicidad de uso.

Visual Paradigm Online
Logotipo  VisualParadigm Online <i>Diagrams</i>
Desarrollador(es) Visual Paradigm International Ltd
Licencia Propietario
Estado Actual Activo

Ilustración 4-9: Características Visual Paradigm Online.



Capítulo

5

ANALISIS DEL SISTEMA

5.1 Introducción

Ante la construcción de cualquier sistema software es necesario realizar un análisis de este, ya que, un mal análisis de los requisitos del sistema afectaría negativamente al mismo. Establecer con exactitud los requisitos de un sistema es, por tanto, un principio esencial para llevar a cabo con éxito un desarrollo software [42].

Así, de acuerdo con las conclusiones obtenidas en el *apartado 3.6* hay que tomar una serie de decisiones sobre los requisitos a cumplir. En este capítulo realizaremos una definición formal del sistema y que requisitos debe cumplir desde la perspectiva del usuario.

5.2 Especificación preliminar del sistema

En forma de resumen de los requisitos y objetivos que el sistema deberá cumplir y que detallaremos y clasificaremos en los siguientes apartados, detallamos los siguientes criterios que la aplicación resultante debe cumplir:

- La aplicación debe permitir la autenticación de usuarios ya existentes, mediante el ingreso de su email y su contraseña.
- La aplicación debe permitir el registro de nuevos usuarios mediante un correo electrónico y una contraseña.
- La autenticación es obligatoria para mostrar el contenido de la aplicación.
- Los cursos podrán ser públicos o privados.
- Los usuarios podrán crear nuevos cursos, nuevas unidades por curso y tarjetas de aprendizaje por cada unidad.
- La aplicación dispondrá de dos categorías de cursos, destinados al aprendizaje de idiomas y destinados a pruebas estandarizadas (pruebas de tipo test).
- Los usuarios deberán estar matriculados en como mínimo un curso público.
- El usuario podrá estar matriculado en varios cursos simultáneamente y debe ser posible cambiar entre ellos, retomando el aprendizaje entre cursos.
- El contenido creado por los usuarios podrá ser compartido con el resto de los usuarios de la aplicación si así lo desean.
- Se permitirá añadir contenido en video a las tarjetas de aprendizaje.
- Se permitirá incorporar una imagen al curso.
- La aplicación utilizará como método de repetición el algoritmo *SuperMemo*.
- La aplicación estará preparada para visualizarse de forma correcta en diferentes tamaños de pantalla.
- El sistema estará diseñado para que los usuarios puedan de una forma sencilla añadir su propio contenido.

Se pretende generar una base de código que permita, tras una compilación específica, aplicaciones para las siguientes plataformas:

- PWA (aplicación web progresiva).
- Aplicación móvil para *Android* e *iOS*.
- Aplicación de escritorio para *Windows* y *Linux*.

5.3 Objetivos del sistema

En esta sección también denominada *características del sistema* (*system features*) o requisitos generales, detallamos los objetivos que se esperan alcanzar cuando el sistema software a desarrollar esté en explotación.

OBJ-001	Gestionar usuarios
Descripción	El sistema deberá gestionar usuarios, permitiendo el registro y la autenticación en el sistema, la modificación de los datos y la baja.
Estabilidad	Alta.
Comentarios	Ninguno.

OBJ-002	Gestionar cursos
Descripción	El sistema deberá gestionar la información correspondiente a los cursos del sistema: altas, bajas y modificación de datos.
Estabilidad	Alta.
Comentarios	Ninguno.

OBJ-003	Gestionar matriculaciones en cursos.
Descripción	El sistema deberá gestionar la información correspondiente a las matriculaciones de usuarios en cursos del sistema: matriculación y desmatriculación.
Estabilidad	Media.
Comentarios	Las posibilidades para matricularse en los cursos pueden variar en versiones posteriores, se establece la estabilidad como media.

OBJ-004	Gestionar unidades
Descripción	El sistema deberá gestionar la información correspondiente a las unidades de los cursos del sistema: altas, bajas y modificación de datos.
Estabilidad	Alta.
Comentarios	Ninguno.

OBJ-005	Gestionar tarjetas de aprendizaje
Descripción	El sistema deberá gestionar la información correspondiente a las tarjetas de aprendizaje de las diferentes unidades: altas, bajas, y modificación de datos.
Estabilidad	Alta.
Comentarios	Ninguno.

OBJ-006	Gestionar sesiones aprendizaje y repaso.
Descripción	El sistema deberá gestionar la información correspondiente a las sesiones de aprendizaje y repaso de las diferentes unidades que contienen los cursos en los que un usuario este matriculado; alta y baja.
Estabilidad	Alta.
Comentarios	Ninguno.

5.4 Requisitos del sistema

Antes de profundizar en los tipos de requisitos, considero preciso definir formalmente lo que se entiende por «requisito software». Según la IEEE⁵⁷, Un requisito software es la capacidad que debe alcanzar o poseer un sistema o componente de un sistema para satisfacer un contrato, estándar, especificación u otro documento formal.

De acuerdo con esta definición, en los consiguientes apartados de esta sección se realizará una clasificación y especificación de los requisitos que debe cumplir el sistema, intentando que sean claros y que cumplan la característica principal de todo requisito software, ser **verificables**, es decir se intentara evitar especificaciones ambiguas, como puede ser “interfaz amigable y fácil de utilizar” ya que esta es de muy difícil verificación al ser difícilmente cuantificable.

Para la documentación de los requisitos, se ha optado por seguir las plantillas propuestas por el **Marco de desarrollo de la Junta de Andalucía (MADEJA)** [43] y atender a las recomendaciones propuestas en el **Informe técnico - Metodología para la Elicitación de Requisitos de Sistemas Software**, propuesto por la universidad de Sevilla en abril de 2002. Dado que se trata de un proyecto con limitaciones, se ha optado por utilizar las plantillas simplificadas propuestas, en la que en la mayoría de los casos se eliminan los campos relativos a versión, autores, fuentes, importancia, urgencia y estado de desarrollo. Se ha optado por mantener el campo Estabilidad, ya que este campo indica la estabilidad del objetivo, es decir una estimación de la probabilidad de que pueda sufrir cambios en el futuro.

5.4.1 Requisitos de información

En este apartado detallaremos los requisitos de almacenamiento y de restricciones de información que deberá tener el sistema. Atendiendo a las recomendaciones de la IEEE y con el objetivo de identificar rápidamente la identificación de los requisitos, se establece el comienzo con las siglas **IRQ** para los **requisitos de almacenamiento** de información y **CRQ** para los **requisitos de restricciones de información**.

IRQ-001	Información sobre Usuarios
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-001 – Gestionar usuarios.
Requisitos asociados	<ul style="list-style-type: none"> • UC-001 – Registrarse • UC-002 - Iniciar Sesión.
Descripción	El sistema deberá almacenar la información de los usuarios. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Identificador de usuario

⁵⁷ **IEEE**: es una asociación mundial de ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas

	<ul style="list-style-type: none"> • Email. • Contraseña. • Imagen de perfil. • Método de registro. • Fecha de registro. • Configuración. • Ultimo acceso. • Cursos Creados • Cursos Matriculados • Curso Actual.
Estabilidad	Alta
Comentarios	Ninguno.

CRQ-001	<i>Unicidad de Usuarios</i>
Objetivos asociados	• OBJ-001 – Gestionar usuarios.
Requisitos asociados	• IRQ-001 – Información sobre usuarios.
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: los usuarios deberán ser únicos, independientemente del método de registro, siendo el campo distintivo el email proporcionado por el usuario. Es decir, un usuario no puede registrarse utilizando como método de acceso email y volver a registrarse desde redes sociales aportando el mismo email. No puede haber dos usuarios con el mismo email.
Comentarios.	Ninguno.

IRQ-002	<i>Información sobre Cursos.</i>
Objetivos asociados	• OBJ-002 – Gestionar Cursos.
Requisitos asociados	<ul style="list-style-type: none"> • UC-009 - Crear cursos. • UC-005 - Visualizar Cursos. • UC-007 - Detalle del curso. • UC-016 - Modificar curso • UC-017 - Eliminar Curso. • UC-026 - Listar todos los cursos.
Descripción	El sistema deberá almacenar la información sobre cursos. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Identificador de curso. • Nombre del curso. • Descripción. • Categoría • Autor. • Fecha de creación. • Estado del curso. • Curso público o privado. • Imagen principal.
Estabilidad	Alta
Comentarios	Ninguno.

IRQ-003	<i>Información sobre Matriculaciones</i>
Objetivos asociados	• OBJ-003 – Gestionar matriculaciones en cursos.
Requisitos asociados	<ul style="list-style-type: none"> • UC-010 - Matricularse de curso. • UC-011 - Desmatricularse de curso.

Descripción	El sistema deberá almacenar la información sobre la matriculación de los usuarios en cursos. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Identificador de usuario. • Identificador de curso. • Fecha de matriculación • Curso principal.
Estabilidad	Alta
Comentarios	Ninguno.

CRQ-003	<i>Unicidad de curso principal</i>
Objetivos asociados	• OBJ-003 – Gestionar matriculaciones en cursos.
Requisitos asociados	• IRQ-003– Información sobre Matriculaciones.
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>los usuarios pueden estar matriculados en más de un curso, pero solo un curso podrá estar establecido como principal.</i>
Comentarios.	Ninguno.

IRQ-004	<i>Información sobre unidades de un curso</i>
Objetivos asociados	• OBJ-004 – Gestionar Unidades.
Requisitos asociados	<ul style="list-style-type: none"> • UC-019 - Crear unidad • UC-018 - Listar unidades • UC-021 - Modificar unidad. • UC-020 - Eliminar unidad
Descripción	El sistema deberá almacenar la información sobre las unidades de un curso. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Identificador de Curso. • Identificador de Unidad. • Número de tarjetas de aprendizaje. • Orden de aprendizaje.
Estabilidad	Alta
Comentarios	Ninguno.

CRQ-004	<i>Unicidad del orden de aprendizaje de unidades del curso.</i>
Objetivos asociados	• OBJ-004 – Gestionar matriculaciones en cursos.
Requisitos asociados	• IRQ-004– Información sobre unidades de un curso.
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: el orden de aprendizaje debe ser único. Es decir, no pueden existir dos unidades con el mismo orden de aprendizaje.
Comentarios.	Ninguno.

IRQ-005	<i>Información sobre tarjetas de aprendizaje de una unidad.</i>
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-005 – Gestionar tarjetas de aprendizaje.
Requisitos asociados	<ul style="list-style-type: none"> • UC-023 - Crear tarjeta de aprendizaje. • UC-022 - Listar tarjetas de aprendizaje. • UC-024 - Eliminar tarjeta de aprendizaje. • UC-025 - Modificar tarjeta de aprendizaje. • UC-027- Practicar tarjetas de aprendizaje de unidad.
Descripción	El sistema deberá almacenar la información sobre las tarjetas de aprendizaje de una unidad. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Identificador de Curso. • Identificador de Unidad. • Identificador de tarjeta de aprendizaje. • Orden de aprendizaje. • Título. • Tipo de tarjeta de aprendizaje. • Datos tarjeta.
Estabilidad	Media.
Comentarios	Los datos específicos pueden variar con la adición de nuevos tipos de tarjeta. Por este motivo se establece la estabilidad como media.

CRQ-005	<i>Unicidad del orden de aprendizaje de tarjetas.</i>
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-005 – Gestionar tarjetas de aprendizaje.
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-005– Información sobre tarjetas de aprendizaje de una unidad.
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: el orden de aprendizaje debe ser único. Es decir, no pueden existir dos tarjetas de aprendizaje con el mismo orden de aprendizaje.
Comentarios.	Ninguno.

CRQ-006	<i>Variabilidad de los datos de tarjeta.</i>
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-005 – Gestionar tarjetas de aprendizaje.
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-005– Información sobre tarjetas de aprendizaje de una unidad.
Descripción	La información almacenada por el sistema deberá satisfacerla siguiente restricción: en función del tipo de tarjeta, los datos almacenados, en el campo datos de tarjeta pueden variar. Es decir, una tarjeta de tipo "traducción", almacenara una frase y su traducción, por el contrario, una tarjeta del tipo "video", almacenara el identificador del video y un fragmento de los sus subtítulos.
Comentarios.	Ninguno.

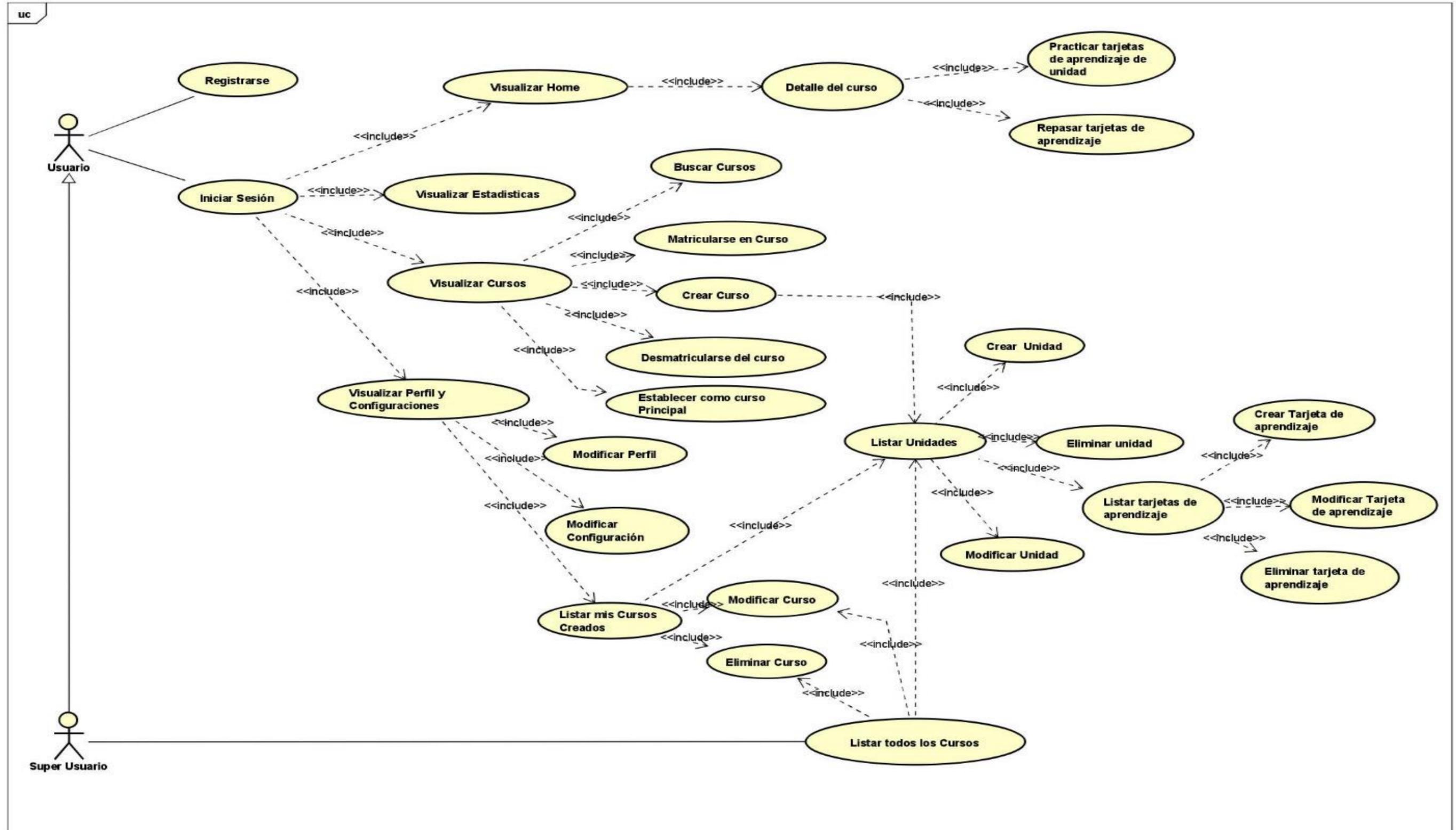
IRQ-006	<i>Información sobre sesiones de aprendizaje y repaso.</i>
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-006 – Gestionar sesiones aprendizaje y repaso.

Requisitos asociados	<ul style="list-style-type: none"> • UC-004 - Visualizar Estadísticas. • UC-028 - Repasar tarjetas de aprendizaje.
Descripción	El sistema deberá almacenar la información sobre las sesiones de aprendizaje y repaso. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Identificador de Curso. • Identificador de Unidad. • Identificador de usuario. • Identificador de tarjeta • Numero de errores. • Numero de aciertos • Fecha de repaso.
Estabilidad	Media.
Comentarios	Los datos específicos pueden variar con la adición de nuevos tipos de tarjeta. Por este motivo se establece la estabilidad como media.

5.4.2 Requisitos funcionales

5.4.2.1 Diagramas de casos de uso

UseCase Diagram



5.4.2.2 Definición de actores

ACT-001	<i>Usuario</i>
Descripción	Este actor de negocio actual representa a un usuario que esta autenticados en el sistema o realizando un registro.
Comentarios	Este usuario puede utilizar los servicios de la aplicación dado que esta autenticado en él sistema. Referente a la creación y modificación de cursos, solo es extensible a los creados por el mismo.

ACT-002	<i>Super Usuario</i>
Descripción	Este actor de negocio actual representa a un usuario que esta autenticados en el sistema.
Comentarios	Este usuario puede utilizar los servicios de la aplicación dado que esta autenticado en el sistema. Además, tiene la posibilidad de realizar operaciones CRUD sobre todos los cursos del sistema.

5.4.2.3 Casos de uso del sistema

UC-001	Registrarse	
Objetivos asociados	• OBJ-001- Gestionar usuarios.	
Requisitos asociados	• IRQ-001 – Información sobre usuarios.	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando algún usuario solicite registro.	
Precondición	El usuario no está registrado en el sistema con el mismo email.	
Secuencia Normal	Paso	Acción
	1	El usuario solicita registrarse en el sistema.
	2	El sistema informa de los tipos de registro y los datos necesarios.
	3	El usuario selecciona el tipo de registro.
	4	El usuario o las pasarelas de registro aportan el email y la contraseña de acceso y solicita el registro.
	5	El sistema almacena la información del usuario e informa al usuario que el registro a se completó con éxito.
	6	Se realiza el caso de uso (UC-002 – Iniciar sesión).
Postcondición	Tiene que cumplirse que el email introducido para la autenticación no puede existir previamente en el sistema.	
Excepciones	Paso	Acción
	4	Si el email introducido, ya existe, el sistema informa al usuario que el email ya está registrado, a continuación, este caso de uso queda sin efecto.
Estabilidad	Alta.	
Comentarios	Ninguno.	

UC-002	Iniciar Sesión.	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-001- Gestionar usuarios. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-001 – Información sobre usuarios. 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando algún usuario solicite Iniciar sesión.	
Precondición	El usuario ha tenido que seleccionar la opción de autenticación.	
Secuencia Normal	Paso	Acción
	1	El usuario solicita iniciar sesión
	2	El sistema informa de los tipos de autenticación disponibles.
	3	El usuario selecciona el tipo de autenticación.
	4	El usuario o las pasarelas de registro aportan el email y la contraseña de acceso y solicita autenticarse.
5	El sistema almacena actualiza la fecha de ultima conexión del usuario e informa al usuario que la autenticación se completó con éxito.	
Postcondición	Tiene que cumplirse que el email introducido corresponda con el de un usuario registrado del sistema y que la contraseña sea correcta para dicho usuario.	
Excepciones	Paso	Acción
	4	Si el email introducido o la contraseña no son correctos el sistema informa al usuario, a continuación, este caso de uso queda sin efecto.
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-003	Visualizar Home.	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-003 - Gestionar matriculaciones en curso. • OBJ-006 - Gestionar sesiones aprendizaje y repaso. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-003 – Información sobre matriculaciones. • IRQ-006 - Información sobre sesiones de aprendizaje y repaso 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando algún usuario solicite visualizar el home.	
Precondición	El usuario ha tenido que autenticarse y seleccionar la opción de Home.	
Secuencia Normal	Paso	Acción
	1	El usuario solicita visualizar el home.
	2	El sistema muestra la información del curso establecido por el usuario como curso principal, con la siguiente información: nombre, descripción, unidad actual en el progreso del curso, porcentajes de progreso.
Postcondición	Ninguna.	
Excepciones	Paso	Acción
	-	-
Estabilidad	Alta.	

Comentarios.	Ninguno.
---------------------	----------

UC-004	Visualizar Estadísticas.	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-003 - Gestionar matriculaciones en curso. • OBJ-006 - Gestionar sesiones aprendizaje y repaso. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-003 – Información sobre matriculaciones. • IRQ-006 - Información sobre sesiones de aprendizaje y repaso 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando algún usuario solicite visualizar estadísticas	
Precondición	El usuario ha tenido que autenticarse y seleccionar la opción de visualizar estadísticas.	
Secuencia Normal	Paso	Acción
	1	El usuario solicita visualizar las estadísticas
	2	El sistema muestra la información del curso establecido por el usuario como curso principal, con la siguiente información: nombre y descripción del curso, porcentajes de progreso y numero de aciertos errores diarios, semanales y mensuales.
Postcondición	Ninguna.	
Excepciones	Paso	Acción
	-	-
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-005	Visualizar Cursos.	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-002 - Gestionar cursos. • OBJ-003 - Gestionar matriculaciones en cursos. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-002 – Información sobre cursos. • IRQ-003 – Información sobre matriculaciones 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando algún usuario solicite visualizar cursos.	
Precondición	El usuario ha tenido que autenticarse y seleccionar la opción de visualizar cursos.	
Secuencia Normal	Paso	Acción
	1	El usuario solicita visualizar cursos.
	2	El sistema muestra un listado de los cursos disponibles del sistema, catalogados como públicos, mostrándose de los datos del curso, nombre, descripción, imagen principal, y si el usuario está matriculado en el curso.
Postcondición	Ninguna.	
Excepciones	Paso	Acción
	-	-
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-006	Visualizar perfil y configuraciones.	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-002 – Gestionar usuarios. 	

Requisitos asociados	<ul style="list-style-type: none"> • IRQ-002 - Información sobre Usuarios • IRQ-003 - Información sobre matriculaciones 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando algún usuario solicite visualizar perfil.	
Precondición	El usuario ha tenido que autenticarse y la opción de visualizar perfil.	
Secuencia Normal	Paso	Acción
	1	El usuario solicita visualizar perfil.
	2	El sistema muestra los siguientes datos del usuario: email, tipo de registro. El sistema muestra un listado de los cursos matriculados o creados por el usuario, mostrando los siguientes datos: nombre del curso y descripción. El sistema muestra un listado de las opciones de configuración establecidas por el usuario.
Postcondición	Ninguna.	
Excepciones	Paso	Acción
	-	-
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-007	<i>Detalle del curso.</i>	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-002 - Gestionar cursos. • OBJ-004 - Gestionar unidades 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-002 - Información sobre cursos. • IRQ-004 - Información sobre unidades de un curso 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando algún usuario solicite visualizar los detalles de un curso.	
Precondición	El usuario ha tenido que autenticarse y seleccionar la opción de visualizar perfil.	
Secuencia Normal	Paso	Acción
	1	El usuario solicita visualizar el detalle de un curso.
	2	El sistema muestra los siguientes datos del curso: nombre, descripción, tipo e imagen. El sistema muestra un listado de las unidades del curso ordenadas por el campo orden, mostrando los siguientes datos: nombre, descripción y número de tarjetas de aprendizaje.
Postcondición	Ninguna.	
Excepciones	Paso	Acción
	-	-
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-008	<i>Buscar cursos.</i>	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-002 - Gestionar cursos. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-002 - Información sobre cursos. 	

Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso: <ul style="list-style-type: none"> • UC-005 – Visualizar Cursos. 	
Precondición	El usuario ha tenido que autenticarse y seleccionar la opción de visualizar cursos, buscar cursos.	
Secuencia Normal	Paso	Acción
	1	El sistema solicita los criterios de búsqueda al usuario. Los datos solicitados serán: nombre del curso, tipo de curso. El usuario confirma los datos de búsqueda.
	2	El sistema muestra un listado de los cursos que cumplen el criterio de búsqueda, catalogados como públicos, mostrándose de los datos del curso, nombre, descripción, imagen principal, y si el usuario está matriculado en el curso.
Postcondición	Ninguna.	
Excepciones	Paso	Acción
	2	El sistema no encuentra ningún curso con los criterios de búsqueda indicados. Se informa al usuario que la búsqueda no tuvo éxito, a continuación, este caso de uso queda sin efecto.
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-009	Crear cursos.	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-002 – Gestionar cursos. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-002 - Información sobre cursos. 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso: <ul style="list-style-type: none"> • UC-005 – Visualizar Cursos. 	
Precondición	El usuario ha tenido que autenticarse y seleccionar la opción de visualizar cursos, crear cursos.	
Secuencia Normal	Paso	Acción
	1	El sistema solicita los siguientes datos del curso al usuario: nombre, descripción, imagen del curso, tipo de curso y si es público o privado. El usuario confirma la creación del curso.
	2	El sistema valida los datos aportados del curso e informa al usuario que el curso fue creado con éxito.
Postcondición	El sistema ha almacenado la información correspondiente al nuevo curso.	
Excepciones	Paso	Acción
	2	Los datos aportados para crear el curso no son válidos, se informa al cliente para que modifique los datos aportados.
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-010	Matricularse de curso.	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-002 – Gestionar cursos. • OBJ-003 - Gestionar matriculaciones en cursos. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-002 - Información sobre cursos. • IRQ-003 - Información sobre Matriculaciones 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso:</p> <ul style="list-style-type: none"> • UC-005 – Visualizar Cursos. 	
Precondición	<ul style="list-style-type: none"> • El usuario ha tenido que autenticarse y seleccionar la opción de visualizar cursos, matricularse en curso. • El usuario no puede estar matriculado en el curso seleccionado. 	
Secuencia Normal	Paso	Acción
	1	El usuario solicita matricularse de un curso.
	2	El sistema valida que el usuario no está matriculado ya en el curso seleccionado y muestra un mensaje de confirmación informando que se matriculo con éxito.
Postcondición	El sistema ha almacenado correctamente la información sobre la matriculación del usuario en el curso.	
Excepciones	Paso	Acción
	-	-
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-011	Desmatricularse de curso.	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-002 – Gestionar cursos. • OBJ-003 - Gestionar matriculaciones en cursos. • OBJ-006 - Gestionar sesiones aprendizaje y repaso. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-002 - Información sobre cursos. • IRQ-003 - Información sobre Matriculaciones • IRQ-006 - Información sobre sesiones de aprendizaje y repaso. 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso:</p> <ul style="list-style-type: none"> • UC-005 – Visualizar Cursos. 	
Precondición	<ul style="list-style-type: none"> • El usuario ha tenido que autenticarse y seleccionar la opción de visualizar cursos, desmatricularse de curso. • El usuario previamente debe estar matriculado en el curso. 	
Secuencia Normal	Paso	Acción
	1	El usuario solicita desmatricularse de un curso.
	2	El sistema muestra un mensaje de confirmación informando que se borrarán los registros de progreso en el curso.
	3	El usuario confirma la desmatriculación del curso.
	4	El sistema elimina la información sobre sesiones de aprendizaje y repaso. El sistema informa al usuario que el proceso termino con éxito.

Postcondición	El sistema ha almacenado correctamente la modificación sobre la matriculación del usuario en el curso y elimino la información sobre las sesiones de aprendizaje y repaso.	
Excepciones	Paso	Acción
	3	Si el usuario tras el mensaje informativo no confirma la desmatriculación, se cancela y este caso de uso queda sin efecto.
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-012	<i>Establecer como curso principal.</i>	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-002 – Gestionar cursos. • OBJ-003 - Gestionar matriculaciones en cursos. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-002 - Información sobre cursos. • IRQ-003 - Información sobre Matriculaciones. 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso: <ul style="list-style-type: none"> • UC-005 – Visualizar Cursos. 	
Precondición	<ul style="list-style-type: none"> • El usuario ha tenido que autenticarse y seleccionar la opción de visualizar cursos, establecer como curso principal. • El usuario previamente debe estar matriculado en el curso. 	
Secuencia Normal	Paso	Acción
	1	El usuario selecciona la opción de establecer el curso como principal.
	2	El sistema informa al usuario que el curso se estableció con éxito como principal.
Postcondición	El sistema ha almacenado correctamente la modificación sobre el curso principal.	
Excepciones	Paso	Acción
	-	-
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-013	<i>Modificar perfil.</i>	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-001 – Gestionar usuarios. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-001 - Información sobre usuarios. 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso: <ul style="list-style-type: none"> • UC-006 - Visualizar perfil y configuraciones. 	
Precondición	El usuario ha tenido que autenticarse y seleccionar la opción de visualizar perfil, modificar datos de perfil.	
Secuencia Normal	Paso	Acción
	1	El sistema muestra al usuario los siguientes datos de usuario: email, contraseña, imagen de perfil. Los campos permitirán la edición.
	2	El usuario selecciona la opción guardar modificaciones.
	3	El sistema valida y almacena la información modificada por el usuario.

	4	El sistema informa al usuario que la información de perfil se modificó correctamente.
Postcondición	El sistema ha almacenado correctamente la modificación sobre el perfil del usuario.	
Excepciones	Paso	Acción
	2	El usuario, cancela el guardado de las modificaciones. Se restablecen los datos anteriores a la modificación en el formulario y este caso de uso queda sin efecto.
	3	En caso de que los datos aportados no sean válidos, se informa al usuario y se regresa al paso 1.
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-014	Modificar Configuración.	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-001 – Gestionar usuarios. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-001 - Información sobre usuarios. 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso:</p> <ul style="list-style-type: none"> • UC-006 - Visualizar perfil y configuraciones. 	
Precondición	El usuario ha tenido que autenticarse y seleccionar la opción de visualizar perfil, modificar configuración.	
Secuencia Normal	Paso	Acción
	1	El sistema muestra al usuario los siguientes datos de la configuración del usuario: <ul style="list-style-type: none"> • Color del tema (claro/oscuro) • Reproducción automática de videos en dispositivos móviles. • ...
	2	El usuario selecciona la opción guardar modificaciones.
	3	El sistema valida y almacena la información modificada por el usuario.
	4	El sistema informa al usuario que la configuración se modificó correctamente.
Postcondición	El sistema ha almacenado correctamente la modificación sobre la configuración.	
Excepciones	Paso	Acción
	2	El usuario, cancela el guardado de las modificaciones. Se restablecen los datos anteriores a la modificación en el formulario y este caso de uso queda sin efecto.
Estabilidad	Baja.	
Comentarios.	Se establece la estabilidad como baja ya que, la configuración posible por parte del usuario dependerá de los tipos de tarjeta de aprendizaje y dispositivos que finalmente se implementen. (Ej: Si se implementan tarjeta de video existirá un parámetro de configuración que permita el auto-play).	

UC-015	Listar mis cursos creados.	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-002 – Gestionar cursos. • OBJ-003 - Gestionar matriculaciones en cursos. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-002 - Información sobre cursos. • IRQ-003 - Información sobre Matriculaciones. 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso:</p> <ul style="list-style-type: none"> • UC-006 - Visualizar perfil y configuraciones. 	
Precondición	<ul style="list-style-type: none"> • El usuario ha tenido que autenticarse y seleccionar la opción de visualizar perfil, listar cursos creados. • El usuario debe ser el autor del curso o SuperUsuario. 	
Secuencia Normal	Paso	Acción
	1	El usuario solicita mostrar los cursos creados por el mismo.
	2	El sistema filtra los cursos que han sido creados por el usuario autenticado y muestra al usuario los siguientes datos de los cursos: nombre, descripción y si es público o privado.
Postcondición	Ninguna.	
Excepciones	Paso	Acción
	2	Si el usuario no tiene cursos creados, el sistema informara al usuario que no dispone de cursos creados por él, a continuación, este caso de uso queda sin efecto.
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-016	Modificar curso.	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-002 – Gestionar cursos. • OBJ-003 - Gestionar matriculaciones en cursos. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-002 - Información sobre cursos. • IRQ-003 - Información sobre Matriculaciones. 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso:</p> <ul style="list-style-type: none"> • UC-015 – Listar mis cursos creados. • UC-026 – Listar todos los cursos. 	
Precondición	<ul style="list-style-type: none"> • El usuario ha tenido que autenticarse y seleccionar la opción de visualizar perfil, listar cursos creados, modificar curso. • El usuario debe ser el autor del curso o SuperUsuario. 	
Secuencia Normal	Paso	Acción
	1	El usuario solicita modificar un curso.
	2	El sistema muestra el curso seleccionado, mostrando los siguientes datos: nombre, descripción imagen del curso. Estos datos serán editables.
	3	El usuario modifica los datos y solicita guardar.
	4	El sistema valida los datos modificados y los almacena.
	5	El sistema notifica al usuario que los datos fueron modificados con éxito.
Postcondición	El sistema ha almacenado correctamente la modificación sobre el curso.	

Excepciones	Paso	Acción
	4	Si los datos del curso modificado no son válidos. El sistema informa al usuario y este caso de uso queda sin efecto.
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-017	<i>Eliminar Curso.</i>	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-002 - Gestionar cursos. • OBJ-003 - Gestionar matriculaciones en cursos. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-002 - Información sobre cursos. • IRQ-003 - Información sobre Matriculaciones. 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso:</p> <ul style="list-style-type: none"> • UC-015 - Listar mis cursos creados. • UC-026- Listar todos los cursos. 	
Precondición	<ul style="list-style-type: none"> • El usuario ha tenido que autenticarse y seleccionar la opción de visualizar perfil, listar cursos creados, eliminar curso. • El usuario debe ser el autor del curso o SuperUsuario. 	
Secuencia Normal	Paso	Acción
	1	El usuario solicita eliminar un curso.
	2	El sistema muestra un mensaje de confirmación informando que la operación es irreversible.
	3	El usuario confirma la eliminación.
	4	El sistema valida que es posible eliminar el curso.
5	El sistema notifica al usuario que el curso fue borrado con éxito.	
Postcondición	El sistema elimino correctamente el curso.	
Excepciones	Paso	Acción
	4	Si el curso es público el curso no puede ser borrado. El sistema informara al usuario y este caso de uso queda sin efecto.
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-018	<i>Listar unidades.</i>	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-004 - Gestionar unidades 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-004 -Información sobre unidades de un curso 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso:</p> <ul style="list-style-type: none"> • UC-015 - Listar mis cursos creados. • UC-009 - Crear cursos. • UC-026 - Listar todos los cursos. 	
Precondición	<ul style="list-style-type: none"> • El usuario ha tenido que autenticarse y seleccionar la opción de visualizar perfil, listar cursos creados, listar unidades de un curso. • El usuario debe ser el autor del curso o SuperUsuario. 	

Secuencia Normal	Paso	Acción
	1	El usuario solicita listar las unidades de un curso.
	2	El sistema muestra un listado con las unidades del curso que contiene los siguientes datos: nombre, descripción y número de tarjetas de aprendizaje.
Postcondición	Ninguna.	
Excepciones	Paso	Acción
	2	Si el curso no dispone de unidades, se muestra un mensaje al usuario y la lista aparecerá vacía.
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-019	Crear unidad.	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-004 - Gestionar unidades 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-004 - Información sobre unidades de un curso 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso:</p> <ul style="list-style-type: none"> • UC-018 – Listar unidades. 	
Precondición	<ul style="list-style-type: none"> • El usuario ha tenido que autenticarse y seleccionar la opción de listar unidades de un curso, crear unidad. • El usuario debe ser el autor del curso o SuperUsuario. 	
Secuencia Normal	Paso	Acción
	1	El usuario solicita añadir una nueva unidad a un curso.
	2	El sistema solicita información sobre la unidad donde se solicitan los siguientes datos: nombre, descripción.
	3	El usuario confirma los datos.
	4	El sistema recibe la información, la valida y asigna como orden de tarjeta el número total de unidades del curso más uno. El sistema guarda la información.
5	El sistema informa al usuario que los datos de la unidad fueron guardados con éxito.	
Postcondición	Ninguna.	
Excepciones	Paso	Acción
	4	Si los datos de la unidad no son válidos, el sistema notificara al usuario volviendo al paso 2.
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-020	Eliminar unidad.	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-004 - Gestionar unidades 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-004 - Información sobre unidades de un curso 	

Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso: <ul style="list-style-type: none"> UC-018 – Listar unidades. 	
Precondición	<ul style="list-style-type: none"> El usuario ha tenido que autenticarse y seleccionar la opción de listar unidades de un curso, eliminar unidad. El usuario debe ser el autor del curso o SuperUsuario. 	
Secuencia Normal	Paso	Acción
	1	El usuario solicita eliminar una nueva unidad a un curso.
	2	El sistema informa que la operación es irreversible al usuario.
	3	El usuario confirma la eliminación de la unidad.
	4	El sistema actualiza el campo orden de todas las unidades con un orden superior. El sistema elimina la unidad.
5	El sistema informa al usuario que la unidad fue borrada correctamente.	
Postcondición	<ul style="list-style-type: none"> El sistema actualiza el orden de las tarjetas con un orden superior para cumplir el requisito CRQ-004 - Unicidad del orden de aprendizaje de unidades del curso. El sistema elimina los datos de la unidad solicitada. 	
Excepciones	Paso	Acción
	4	Si la unidad contiene tarjetas de aprendizaje, el sistema informa al usuario que no es posible eliminar, la unidad mientras existan tarjetas de aprendizaje y este caso de uso queda sin efecto.
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-021	Modificar unidad.	
Objetivos asociados	<ul style="list-style-type: none"> OBJ-004 - Gestionar unidades 	
Requisitos asociados	<ul style="list-style-type: none"> IRQ-004 - Información sobre unidades de un curso 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso: <ul style="list-style-type: none"> UC-018 – Listar unidades. 	
Precondición	<ul style="list-style-type: none"> El usuario ha tenido que autenticarse y seleccionar la opción de listar unidades de un curso, modificar unidad. El usuario debe ser el autor del curso o SuperUsuario. 	
Secuencia Normal	Paso	Acción
	1	El usuario solicita modificar la unidad.
	2	El sistema permite editar los siguientes datos de la unidad: nombre, descripción y orden.
	3	El usuario confirma guardar las modificaciones de la unidad.
4	El sistema almacena las modificaciones de la unidad.	

	5	El sistema informa al usuario que la unidad modificada se modificó correctamente.
Postcondición	El sistema modifica los datos de la unidad solicitada.	
Excepciones	Paso	Acción
	4	Si las modificaciones no son válidas para almacenarlas en el sistema, el sistema informa al usuario y este caso de uso queda sin efecto.
	4	Si es modificado el campo orden, el sistema actualiza el orden de todas las tarjetas del curso para cumplir el requisito CRQ-004 - Unicidad del orden de aprendizaje de unidades del curso.
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-022	Listar tarjetas de aprendizaje.	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-005 – Gestionar tarjetas de aprendizaje 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-005 - Información sobre tarjetas de aprendizaje de una unidad. 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso:</p> <ul style="list-style-type: none"> • UC-018 – Listar unidades. 	
Precondición	<ul style="list-style-type: none"> • El usuario ha tenido que autenticarse y seleccionar la opción de listar unidades de un curso, listar tarjetas de aprendizaje. • El usuario debe ser el autor del curso o SuperUsuario. 	
Secuencia Normal	Paso	Acción
	1	El usuario solicita listar las tarjetas de aprendizaje de una unidad.
	2	El sistema lista las tarjetas de aprendizaje ordenadas por el campo orden y mostrando los siguientes datos: título, tipo y datos.
Postcondición	Ninguna.	
Excepciones	Paso	Acción
	2	Si la unidad no tiene tarjetas de aprendizaje el sistema notifica al usuario y este caso de uso queda sin efecto.
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-023	Crear tarjeta de aprendizaje.	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-005 – Gestionar tarjetas de aprendizaje 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-005 - Información sobre tarjetas de aprendizaje de una unidad. 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso:</p> <ul style="list-style-type: none"> • UC-022 – Listar tarjetas de aprendizaje. 	
Precondición	<ul style="list-style-type: none"> • El usuario ha tenido que autenticarse y seleccionar la opción de listar unidades de un curso, crear tarjeta de aprendizaje. • El usuario debe ser el autor del curso o SuperUsuario. 	

Secuencia Normal	Paso	Acción
	1	El usuario solicita crear una nueva tarjeta de aprendizaje.
	2	El sistema muestra los tipos de tarjeta de aprendizaje.
	3	El usuario decide qué tipo de tarjeta desea crear.
	4	El sistema solicita la información necesaria para crear el tipo de tarjeta.
	5	El usuario aporta la información dependiendo del tipo de tarjeta y solicita guardar la tarjeta.
	6	El sistema recibe la información, la valida y asigna como orden de tarjeta el número de tarjetas de la unidad más uno. El sistema guarda la información.
	7	El sistema informa al usuario que la tarjeta se almaceno con éxito.
Postcondición	El sistema almacena correctamente la tarjeta de aprendizaje.	
Excepciones	Paso	Acción
	6	Si la unidad no tiene tarjetas, se asigna como orden de tarjeta uno.
	6	Si los datos de la tarjeta no son válidos, el sistema informa al usuario permitiendo continuar desde el paso 4.
Estabilidad	Media.	
Comentarios.	La estabilidad de este caso de uso es media, ya que, los datos de la tarjeta de aprendizaje variaran según el tipo de tarjeta.	

UC-024	<i>Eliminar tarjeta de aprendizaje.</i>	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-005 – Gestionar tarjetas de aprendizaje 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-005 - Información sobre tarjetas de aprendizaje de una unidad. 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso: <ul style="list-style-type: none"> • UC-022 – Listar tarjetas de aprendizaje. 	
Precondición	<ul style="list-style-type: none"> • El usuario ha tenido que autenticarse y seleccionar la opción de listar unidades de un curso, eliminar tarjetas de aprendizaje. • El usuario debe ser el autor del curso o SuperUsuario. 	
Secuencia Normal	Paso	Acción
	1	El usuario solicita eliminar una tarjeta de aprendizaje.
	2	El sistema informa que la operación es irreversible.
	3	El usuario confirma el borrado.
	4	El sistema modifica el orden de todas las tarjetas con orden superior a la tarjeta eliminada, bajando el orden una unidad.
	5	El sistema elimina la tarjeta modificada e informa al usuario que la tarjeta fue eliminada correctamente.

Postcondición	<ul style="list-style-type: none"> • El sistema actualiza el orden de las tarjetas con un orden superior para cumplir el requisito CRQ-005 - Unicidad del orden de aprendizaje de tarjetas. • El sistema elimina la tarjeta de aprendizaje solicitada. 	
Excepciones	Paso	Acción
	-	-
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-025	<i>Modificar tarjeta de aprendizaje.</i>	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-005 – Gestionar tarjetas de aprendizaje 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-005 - Información sobre tarjetas de aprendizaje de una unidad. 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso:</p> <ul style="list-style-type: none"> • UC-022 – Listar tarjetas de aprendizaje. 	
Precondición	<ul style="list-style-type: none"> • El usuario ha tenido que autenticarse y seleccionar la opción de listar unidades de un curso, modificar tarjeta de aprendizaje. • El usuario debe ser el autor del curso o SuperUsuario. 	
Secuencia Normal	Paso	Acción
	1	El usuario solicita modificar una tarjeta de aprendizaje.
	2	El sistema permite editar los campos de la tarjeta de aprendizaje. Los datos que se permiten editar son: datos de tarjeta y orden.
	3	El usuario confirma las modificaciones.
	4	El sistema guarda las efectuadas sobre la tarjeta e informa al usuario que las modificaciones fueron realizadas con éxito.
Postcondición	El sistema almacena la modificación de la tarjeta de aprendizaje.	
Excepciones	Paso	Acción
	4	Si el orden de la tarjeta fue modificado, el sistema modifica el orden de todas las tarjetas para cumplir la restricción CRQ-005 - Unicidad del orden de aprendizaje de tarjetas
Estabilidad	Media.	
Comentarios.	La estabilidad de este caso de uso es media, ya que, los datos de la tarjeta de aprendizaje que son permitidos editar variaran según el tipo de tarjeta.	

UC-026	<i>Listar todos los cursos.</i>	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-002 – Gestionar cursos. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-002 - Información sobre cursos. 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso:</p>	
Precondición	El usuario debe estar autenticado y debe ser SuperUsuario.	
	Paso	Acción

Secuencia Normal	1	El usuario solicita visualizar todos los cursos.
	2	El sistema muestra un listado de los cursos con los siguientes datos: nombre del curso, descripción, autor del curso.
Postcondición	Ninguna.	
Excepciones	Paso	Acción
	-	-
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-027	<i>Practicar tarjetas de aprendizaje de unidad.</i>	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-005 – Gestionar tarjetas de aprendizaje • OBJ-006 - Gestionar sesiones aprendizaje y repaso. 	
Requisitos asociados	<ul style="list-style-type: none"> • IRQ-005 - Información sobre tarjetas de aprendizaje de una unidad. • IRQ-006 - Información sobre sesiones de aprendizaje y repaso. 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso:</p> <ul style="list-style-type: none"> • UC-007 – Detalle del curso. 	
Precondición	<ul style="list-style-type: none"> • El usuario ha tenido que autenticarse y seleccionar la opción de listar unidades de un curso, practicar tarjetas de aprendizaje de unidad. 	
Secuencia Normal	Paso	Acción
	1	El usuario solicita practicar las tarjetas de aprendizaje de una unidad.
	2	El sistema obtiene las tarjetas de aprendizaje de la unidad y las muestra al usuario individualmente y por el orden establecido.
	3	El usuario completa individualmente cada tarjeta de aprendizaje, hasta que el número de tarjetas de aprendizaje pendientes de completar sea cero.
	4	El sistema guarda las informaciones sobre el progreso en cada tarjeta de aprendizaje y en función de los datos de errores y aciertos, programa la repetición de las tarjetas.
	5	El sistema informa al usuario que las tarjetas de aprendizaje pendientes de la unidad se han completado.
Postcondición	El sistema almacena la fecha de repetición y los datos de progreso de la tarjeta de aprendizaje.	
Excepciones	Paso	Acción
	4	Si una unidad no contiene tarjetas de aprendizaje, el sistema informara al usuario y este caso de uso quedara sin efecto.
Estabilidad	Alta.	
Comentarios.	Ninguno.	

UC-028	<i>Repasar tarjetas de aprendizaje.</i>	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-005 – Gestionar tarjetas de aprendizaje • OBJ-006 - Gestionar sesiones aprendizaje y repaso. 	

Requisitos asociados	<ul style="list-style-type: none"> • IRQ-005 - Información sobre tarjetas de aprendizaje de una unidad. • IRQ-006 - Información sobre sesiones de aprendizaje y repaso. 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso durante la realización del caso de uso:</p> <ul style="list-style-type: none"> • UC-007 – Detalle del curso. 	
Precondición	<ul style="list-style-type: none"> • El usuario ha tenido que autenticarse y seleccionar la opción detalle del curso, repasar tarjetas de aprendizaje. 	
Secuencia Normal	Paso	Acción
	1	El usuario solicita repasar las tarjetas de aprendizaje.
	2	El sistema comprueba si existen tarjetas de aprendizaje programadas con fecha superior a la actual. El sistema muestra las tarjetas de aprendizaje al usuario. El sistema asigna un orden, a cada tarjeta de aprendizaje.
	3	El usuario completa individualmente cada tarjeta de aprendizaje.
	4	El sistema guarda las informaciones sobre el progreso en cada tarjeta de aprendizaje y en función de los datos de errores y aciertos, programa la repetición de las tarjetas.
	5	El sistema informa al usuario que las tarjetas de aprendizaje pendientes de repaso han sido completadas.
Postcondición	El sistema almacena la fecha de repetición y los datos de progreso de la tarjeta de aprendizaje.	
Excepciones	Paso	Acción
	4	Si no existen tarjetas de aprendizaje con fecha excedida, el sistema informara al usuario y este caso de uso quedara sin efecto.
Estabilidad	Alta.	
Comentarios.	Ninguno.	

5.4.3 Requisitos no funcionales

RNF-001	Compatibilidad
Descripción	El sistema deberá desarrollarse para su instalación en plataformas Android, iOS, Windows y Linux. El sistema deberá visualizarse en navegadores como una (PWA).
Comentarios	Se utilizará Cordova para generar la aplicación Android e iOS y Electron para generar las compilaciones de Linux y Windows.

RNF-002	Integridad
Descripción	El sistema deberá estar protegido para que usuarios no autorizados no puedan añadir, borrar o modificar datos.
Comentarios	Se definirán reglas de seguridad en Cloud Firestore.

RNF-003	Interfaces de usuario
Descripción	El sistema deberá estar formado por interfaces de usuario adaptadas a los diferentes tipos de pantalla (móviles, Tablet y ordenador).
Comentarios	Se utilizarán diseños adaptativos al tamaño y formato de la pantalla en la que se visualicen (RWD).

5.4.4 Matrices de trazabilidad

En esta sección se presentarán las diferentes matrices de trazabilidad o rastreabilidad con el objetivo de que se pueda conocer con que requisito está asociado cada objetivo. Las matrices de rastreabilidad que se presentan son las siguientes:

- Matriz de trazabilidad de **Requisitos de información** frente a **Objetivos generales** (ver Tabla 5-1).
- Matriz de trazabilidad de **Casos de Uso** frente a **Objetivos Generales** (ver Tabla 5-2).
- Matriz de trazabilidad de **Casos de Uso** frente a **Requisitos de información** (ver Tabla 5-3).

	IRQ 001	IRQ 002	IRQ 003	IRQ 004	IRQ 005	IRQ 006
IRQ-001	•					
IRQ-002		•				
IRQ-003			•			
IRQ-004				•		
IRQ-005					•	
IRQ-006						•

Tabla 5-1: Matriz de trazabilidad de requisitos de información frente a objetivos generales

	OBJ 001	OBJ 002	OBJ 003	OBJ 004	OBJ 005	OBJ 006
UC-001	•					
UC-002	•					
UC-003			•			•
UC-004			•			•
UC-005		•	•			
UC-006	•					
UC-007		•		•		
UC-008		•				
UC-009		•				
UC-010		•	•			
UC-011		•	•			•
UC-012		•	•			
UC-013	•					
UC-014	•					
UC-015		•	•			
UC-016		•	•			
UC-017						
UC-018				•		
UC-019				•		
UC-020				•		
UC-021				•		
UC-022					•	
UC-023					•	
UC-024					•	
UC-025					•	
UC-026		•				
UC-027					•	•
UC-028					•	•

Tabla 5-2: Matriz de trazabilidad de casos de uso frente a objetivos Generales

	IRQ 001	IRQ 002	IRQ 003	IRQ 004	IRQ 005	IRQ 006
UC-001	•					
UC-002	•					
UC-003			•			•
UC-004			•			•
UC-005		•	•			
UC-006	•		•			
UC-007		•		•		
UC-008		•				
UC-009		•				
UC-010		•	•			•
UC-011		•	•			•
UC-012		•	•			
UC-013	•					
UC-014	•					
UC-015		•	•			
UC-016		•	•			
UC-017						
UC-018				•		
UC-019				•		
UC-020				•		
UC-021				•		
UC-022					•	
UC-023					•	
UC-024					•	
UC-025					•	
UC-026		•				
UC-027					•	•
UC-028					•	•

Tabla 5-3: Matriz de trazabilidad de casos de uso frente a requisitos de información



Capítulo

6

DISEÑO Y CONSTRUCCION DEL SISTEMA

6.1 Introducción

Diseñar y construir, no es solo añadir líneas de código al sistema que den como resultado una aplicación, es de gran importancia disponer de una correcta estructuración del proyecto, ya que el software se escribe una vez, pero se lee decenas, cientos de veces, y cuanto mayor sea el número de **buenas prácticas** empleadas en la construcción, más fácil de entender y mantener será [42].

En base a los requisitos analizados en el *capítulo 5*, se pretende obtener una descripción de la mejor solución software que de soporte a dichos requisitos, para esto no solamente se tendrán en cuenta aspectos técnicos, sino también aspectos como los plazos de desarrollo establecidos y aspectos de calidad. En este capítulo se decidirá cuál es la arquitectura del sistema y el diseño de la base de datos y se realizará un diseño de las principales pantallas de la interfaz del usuario, con el objetivo de representar la estructura de la aplicación y como estará organizado el contenido en el sitio.

Además, se realizará un repaso por el modo de construcción de la aplicación, analizando las diferentes técnicas empleadas para la construcción y como utilicé las herramientas y tecnologías descritas en capítulos anteriores, para la construcción mantenible de la aplicación resultante.

6.2 Arquitectura del sistema

Como se definió en el *capítulo 1*, el objetivo es la construcción de una aplicación que con la **misma base de código** permita generar versiones para distintas plataformas. Para conseguir esto se ha optado por hacer uso de la arquitectura que representamos en la *Ilustración 6-1* donde, existe una separación entre la aplicación y la plataforma de utilización.



Ilustración 6-1: Arquitectura global de la APP

Esta separación es posible gracias a la compilación específica para cada plataforma, ya que como podemos apreciar existe un *Wrapper* o envoltorio nativo que soporta la aplicación para cada una de las plataformas o sistemas operativos. Además, como podemos apreciar en la anterior ilustración, existe una parte común independiente de la plataforma de uso, esta parte se encuentra identificada por el color azul oscuro en la ilustración y contiene toda la funcionalidad de la APP. Se ha decidido representar esta parte mediante un único bloque con el objetivo de reflejar la **base de código común** existente e **independiente de la plataforma** final de uso. Podemos apreciar en la *Ilustración 6-2* una representación más detallada de esta parte común y su composición interna.

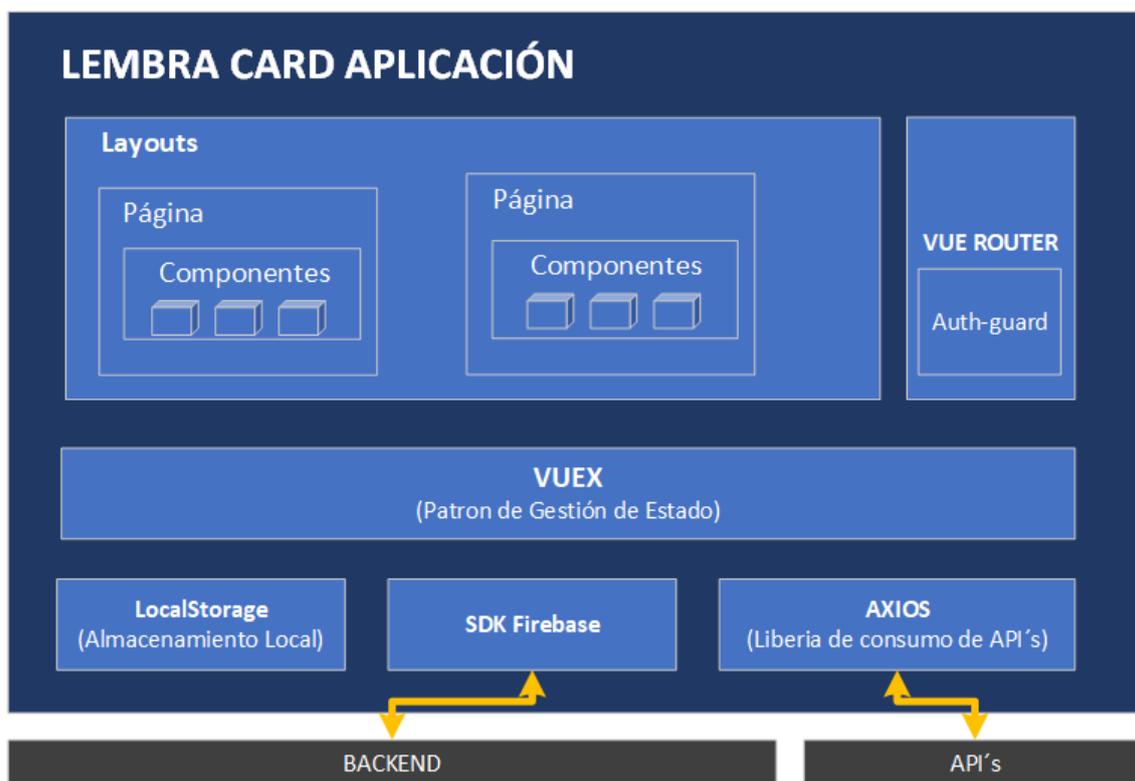


Ilustración 6-2: Arquitectura de la APP Lembra

Describiremos la ilustración anterior de la arquitectura comenzando por la parte inferior, donde ubicamos los sistemas externos que dan soporte a la aplicación, en primer lugar, tenemos el *Backend* que en nuestro caso es *Firebase* y de él que ya hablamos en la *sección 4.6*, en él se aloja la base de datos *Cloud Firestore*. Por otro lado, tenemos las *API's* de proveedores externos a las que realizamos conexiones para obtener datos, algunas de estas son *Pixabay* o *Youtube* y de las que ya hablamos en la *sección 4.8*.

Dentro de la zona delimitada por el color azul, disponemos en un primer nivel en el que se encuentran los sistemas que interactúan con los datos de la aplicación. Por un lado, tenemos el *LocalStorage*, el cual nos permite realizar el almacenamiento local de ciertos datos en la aplicación. Por otro lado, disponemos del *SDK de Firebase* el cual nos permite interactuar con el *Backend* y del que ya hablamos en la *sección 4.6*. Por último, dentro de esta capa o nivel disponemos de *Axios* la cual es una librería de *JavaScript* que nos permite consumir servicios de *API's* y de la que hablamos en la *sección 4.5.2*.

En el siguiente nivel disponemos de *Vuex* que nos permite gestionar los valores de las variables/objetos de la aplicación de una forma centralizada, explicaremos este en detalle en la *sección 6.4*.

En un nivel superior, disponemos por un lado de *Vue Router* quien nos permite gestionar el acceso a las diferentes páginas que conforman la aplicación, además también es el encargado de controlar la autorización de acceso mediante el componente *auth-guard*. Explicaremos su funcionamiento en la *sección 6.4*.

En el nivel superior disponemos de una agrupación compuesta por *layouts*, páginas y componentes. Podemos definir los *layouts* en nuestra aplicación como esquemas de distribución de páginas diferenciados de esta para su reutilización. Además, dentro de las páginas, disponemos de los componentes, representados para comprender de forma más detallada en la *Ilustración 6-3*, donde podemos apreciar los diferentes componentes y como esta arquitectura de construcción de la aplicación favorece su reutilización. Veremos esto en más detalle en la *sección 6.3*

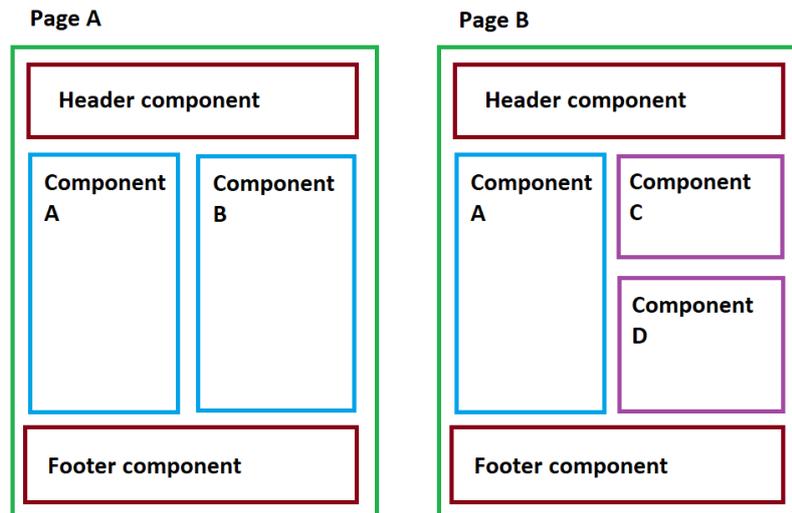


Ilustración 6-3: Representación de componentes páginas y layout

Por ultimo y para comprender de una forma más visual como es el comportamiento de la aplicación cuando se realiza una petición de una página, podemos apreciar la *Ilustración 6-4* en la que se representa el flujo de comunicación y los elementos más importantes que intervienen y que explicamos en este capítulo.

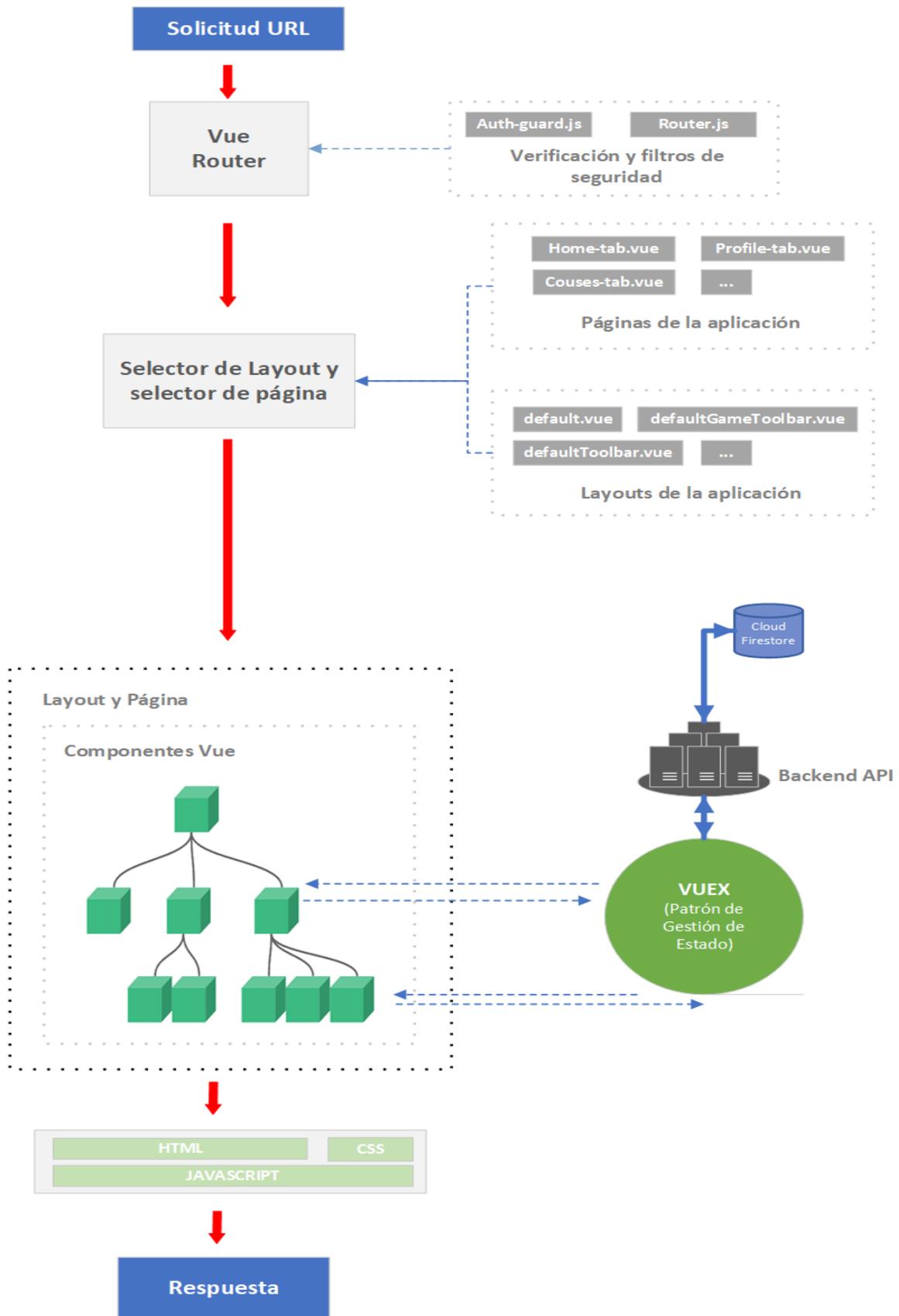


Ilustración 6-4: Esquema de una petición

6.3 Componentes de un solo archivo

Uno de los conceptos que me parecieron interesantes en las primeras etapas de toma de contacto con *Vue*, son los componentes de un solo archivo o *Single File Components (SFCs)*. Como su propio nombre indica, son archivos que contienen un conjunto de diferentes lenguajes y tecnologías que permiten crear elementos personalizados **reutilizables** y que mantienen su **funcionalidad encapsulada** lejos del resto del código [44].

La *W3C*⁵⁸, encargada de generar las recomendaciones y estándares de internet, ya refleja su interés por los componentes web, aunque aún, estos no son un estándar definido y es por esto que no todos los navegadores soportan esta tecnología, a la fecha de la realización de este trabajo fin de grado el navegador web *Mozilla Firefox*⁵⁹ ya incluye la posibilidad de habilitar los componentes en el navegador de modo experimental [45].

Los componentes han hecho que nuestra aplicación sea una aplicación modular. Una visión clara de la modularidad resultante de la aplicación podemos verla en la *Ilustración 6-5* que refleja, a la izquierda cualquiera de las vistas de nuestra aplicación y como, se encuentra estructurada de una forma modularizada. Cada uno de estos componentes cuelgan de forma jerárquica de otros, hasta llegar al elemento raíz de la jerarquía que en nuestro caso son las diferentes páginas de la aplicación.

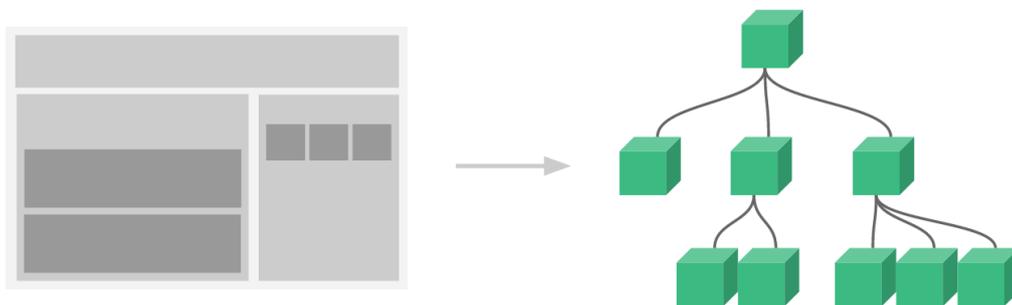


Ilustración 6-5: Componentes en la interfaz

Para comprender en profundidad el concepto de componente, y como hacemos uso de ellos para la construcción, podemos apreciar la *Ilustración 6-6*, en esta vemos un fragmento de código de un componente de nuestra aplicación, que como podemos apreciar se encuentra estructurado en tres

⁵⁸ **W3C**: World Wide Web Consortium (W3C), es un consorcio internacional que genera recomendaciones y estándares que aseguran el crecimiento de la World Wide Web a largo plazo.

⁵⁹ **Mozilla Firefox**: es un navegador web libre y de código abierto desarrollado para Linux, Android, IOS, OS X y Microsoft Windows coordinado por la Corporación Mozilla.

bloques diferenciados: una sección *template*, una sección *script* y la sección *style*. Esta forma de **estructuración modular** puede parecer desestructurada, ya que, en los desarrollos web tradicionales, existía una separación entre los archivos *CSS*, *JavaScript* y *HTML*. Por el contrario, en los componentes, conviven los tres lenguajes en un mismo archivo y como hemos dicho, parece desestructurado, pero no lo es, ya que la separación de lenguajes está claramente diferenciada en secciones dentro del mismo archivo.

```

<! ----- Template -----> #TEMPLATE
<template>
  <div class="progress-bar" :style=
  "{percentage + '%'}">
  </div>
</template> # FIN TEMPLATE

<! ----- Script -----> # SCRIPT
<script>
  export default {
    name: 'CcHeader',
    props: ['cardCount', 'maxCards']
    percentage() {
      return cardCount / maxCards
* 100
    }
  }
</script> # FIN SCRIPT

<! ----- Style -----> # STYLE
<style>
.progress-bar {
  width: 100%
}
</style> # FIN STYLE

```

Ilustración 6-6: Componente CCHheader

El componente que podemos apreciar en la *Ilustración 6-6*, se corresponde con la barra de progreso que aparece en las tarjetas de aprendizaje, es uno de los componentes más simples de nuestra aplicación. Recibe dos propiedades, el número de tarjeta actual y el número total de tarjetas, mediante estas dos propiedades calcula el porcentaje de progreso completado, mostrando como resultado la barra superior que muestra el avance en la sesión de aprendizaje, podemos apreciar el resultado del componente en la *Ilustración 6-7* donde se aprecia el componente en dos tarjetas de aprendizaje diferentes.

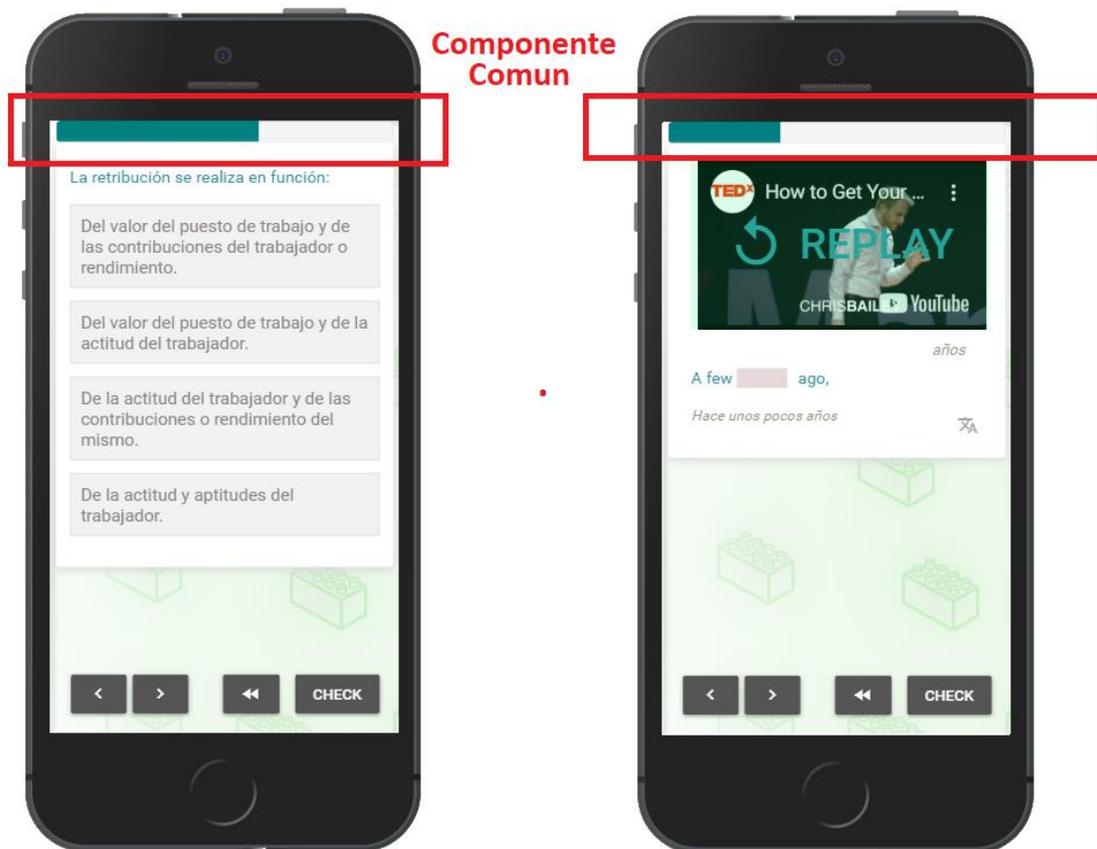


Ilustración 6-7: Vista del componente CcHeader

La funcionalidad para la que fue creado el componente se encuentra encapsulada lejos del resto del código, siendo **fácilmente reutilizable** en esta misma aplicación u otras. El uso de este componente que estamos utilizando como ejemplo se realiza desde otro componente padre, haciéndose uso del mismo como podemos ver en la Ilustración 6-8.

```

<template>
  ...
  <CcHeader
  :cardCount="completedCards.length" :maxCards="maxCards">
  </CcHeader>
  ...
  ...
</template>
<script>
import CcHeader from './Cards/CcHeader.vue'
...
...
</script>
    
```

Ilustración 6-8: Utilización del componente CcHeader

Como podemos apreciar en la anterior ilustración, tras la importación del componente, nos permite hacer uso del mismo de una forma sencilla, pasándole las propiedades que necesita para realizar su funcionalidad de forma **aislada de resto de código**. Por último, indicar que, la idea de modularidad hace que los componentes sean archivos con pocas líneas de código, fácilmente mantenibles.

6.4 Patrón de gestión de estado

En el *apartado 6.3* de este capítulo hemos hablado sobre los componentes que conforman nuestra aplicación, la modularidad que ofrecen, nos permite **reusabilidad** y **encapsulamiento de responsabilidad**, pero el uso de numerosos componentes no son solo ventajas, ya que la comunicación y el paso de estados a través de múltiples componentes se vuelve difícil. Para solventar los problemas ocasionados del uso de componentes, una de las técnicas recomendadas y empleadas por la comunidad *Vue.js* es el uso de un patrón de estado o tienda.

Antes de continuar debemos tener claro que un estado puede verse como el valor actual de una variable/objeto dentro de un componente o una aplicación. Si consideramos que nuestra aplicación está compuesta por numerosos componentes, nos encontramos con la problemática de que una misma variable/objeto pueda estar definida en la aplicación con dos valores distintos, preguntándonos en este caso ¿Cuál es el valor actual o correcto? Sabido esto, podemos definir un estado a nivel de componente y un estado a nivel de aplicación. El **estado a nivel de componente** se puede definir como un estado limitado a un componente. El **estado a nivel de aplicación** es similar, pero a menudo se usa en múltiples componentes o servicios.

Para facilitar la comunicación entre componentes y garantizar que el estado de una variable/objeto es el esperado y que solo se puede mutar de manera predecible, hacemos uso de lo que se conoce como **patrón de tienda** o **patrón de estado**, cuya biblioteca para aplicaciones *Vue.js* es conocida con el nombre de *Vuex* y de la que hablamos en el *apartado 4.5.2.3.2*.

Un ejemplo de cómo encaja la tienda *Vuex* en nuestra aplicación, podemos verla en la *Ilustración 6-9*.

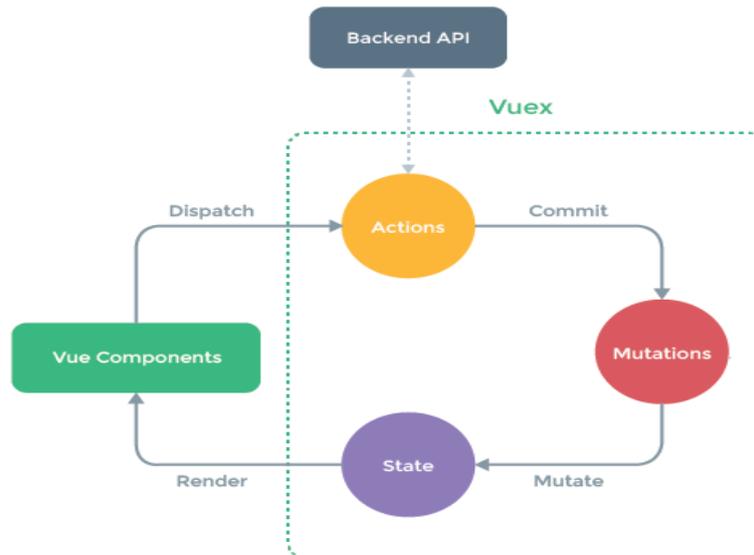


Ilustración 6-9: Flujo Estado Vuex

Para comprender la ilustración anterior, definiremos el proceso paso a paso en un cambio de estado de una variable/objeto de nuestra aplicación.

1. El estado de una variable/objeto se representa en un componente.
2. Un componente envía una *Actions* para obtener datos de la *API Backend*.
3. Se dispara un evento *commit* que es manejado por una *mutation*. Esta mutación devuelve una nueva versión del estado con los datos de respuesta de la llamada a la *API Backend*.
4. El nuevo estado se renderiza en el/los componentes *Vue*.

Por lo tanto, la tienda *Vuex*, es en nuestra aplicación la **única fuente de verdad** en todos los componentes que requieren de un estado a nivel de aplicación. Este diseño nos permite agregar nuevas funciones a nuestra aplicación de manera consistente. [46]

6.5 Patrón Enrutador

Hasta hace unos años era común que las aplicaciones web existentes, constaran de diferentes páginas, cada una de ellas vinculada con una ruta que se activaba cuando la dirección coincidía, lo que podía dar una experiencia de navegación poco fluida. Como definimos en el *apartado 1.3*, la aplicación que se pretende construir permitirá su utilización en diferentes plataformas y dispositivos, por lo que, un enrutamiento tradicional en el que, cada ruta se vincule a una página, no ofrecería una experiencia de navegación óptima en las aplicaciones de escritorio o móvil. Para solventar esta problemática, se ha optado por realizar un desarrollo de la aplicación como una **SPA** (*Single Page Application*), que como podemos definir del propio nombre, son las aplicaciones *que se ejecutan en una sola página* con el propósito de

dar una experiencia más fluida a los usuarios como una aplicación de escritorio.

Sabido que la construcción de la aplicación se ha realizado como una SPA, podemos pensar que no existen rutas al disponer de una única página, pero esto no es cierto, las paginas existen, pero estas junto con los componentes, se cargan de forma dinámica, añadiéndose normalmente en respuesta a las acciones del usuario por lo que se hace necesario una **gestión de las rutas de la aplicación**.

Por este motivo y haciendo uso de las recomendaciones de la comunidad de *Vue.js*, en la construcción de la aplicación hacemos uso de *vue-router* como patrón de enrutamiento. La implementación del enrutamiento con esta biblioteca se ha realizado mediante la definición de **rutas nombradas**. Podemos ver un ejemplo de una de estas rutas en la *Tabla 6-1*, donde como vemos se encuentra definida la ruta *"/game"* la cual, al ser referenciada importa la página *PGame* con el *layout* *defaultGameToolbar*.

```
path: '/game',
  component: () =>
import('layouts/defaultGameToolbar'),

  children: [{
    path: '/game',
    component: () => import('pages/PGame'),
    beforeEnter: AuthGuard
    props: true
  ]
}]
```

Tabla 6-1: Ejemplo de enrutamiento de la App

Por último, en la ilustración anterior, es de interés la propiedad *beforeEnter*, esta propiedad llama a un **guarda de navegación** que como su propio nombre indica se utiliza para proteger la navegación a rutas no autorizadas, ya sea redirigiéndolas o cancelándolas si fuese necesario.

6.6 El modelo de clases

En la *Ilustración 6-10* representamos el *diagrama de clases UML* correspondiente al modelo estático del sistema, mostrándose las principales clases y las relaciones entre ellas, pero sin detallar operaciones ni atributos.

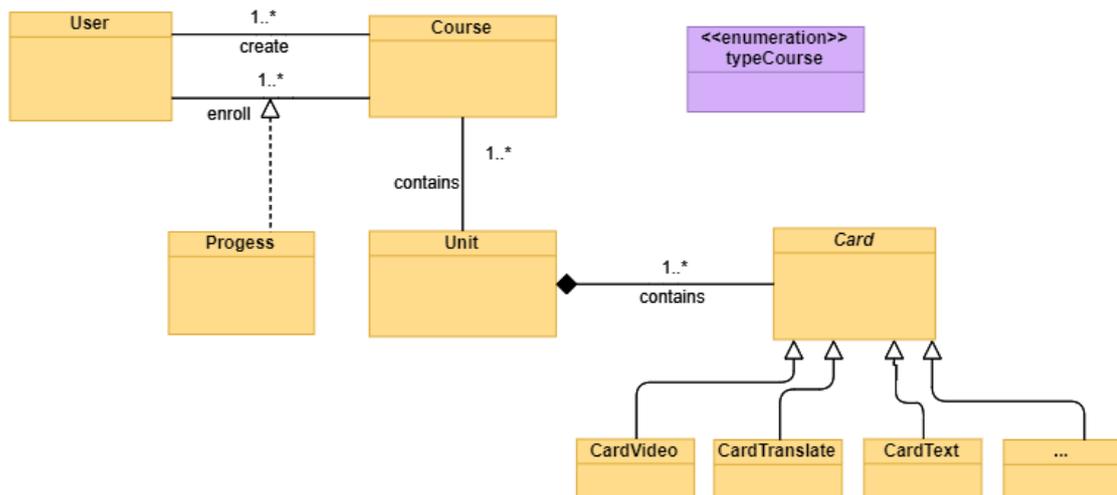


Ilustración 6-10: Diagrama de clases

En la ilustración anterior podemos apreciar que existen varias relaciones de composición, es decir un objeto como por ejemplo cursos, está compuesto por una o varias unidades y a su vez estas por varias tarjetas. Esta relación de composición indica que el tiempo de vida de los objetos de tipo tarjeta dependen del tipo unidad, ya que si no existe una unidad no pueden existir tarjetas para la unidad.

Por otro lado, en la ilustración anterior podemos observar que existe la clase asociación⁶⁰ denominada *Progress* fruto de la asociación que se produce cuando un usuario se matricula en un curso, esta permite representar la necesidad de almacenar el progreso de un usuario en un determinado curso matriculado.

Por último, haciendo referencia con la anterior ilustración comentar la representación de la clase abstracta⁶¹ *Card*, representada en notación UML⁶² con letra cursiva y sobre la cual existen diferentes especializaciones para las diferentes tarjetas de los diferentes tipos.

⁶⁰ **Clase asociación:** en un diagrama de clases, las clases asociativas se derivan de una asociación entre dos o más clases. Una instancia de esta clase está asociada siempre a una y solo una instancia de cada participante de la asociación.

⁶¹ **Clase abstracta:** en un diagrama de clases son utilizadas cuando se desea definir una abstracción que englobe objetos de distintos tipos. Excepto la capacidad de instanciación, tienen las mismas capacidades que una clase concreta.

⁶² **UML:** es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación.

6.7 El Modelo de almacenamiento de datos

6.7.1 Introducción

Como describimos en la *sección 4.7.2*, la base de datos utilizada para la aplicación es *Cloud Firestore*, en este tipo de base de datos, la unidad de almacenamiento es el documento, y a su vez, estos viven en colecciones que simplemente podemos definir como contenedores de documentos, esta jerarquía podemos apreciarla en la *Ilustración 6-11*.

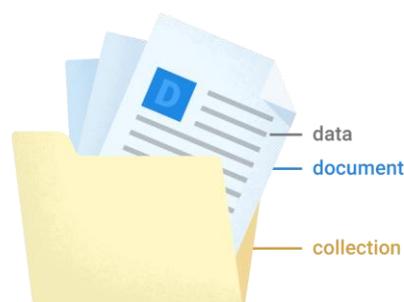


Ilustración 6-11: Modelo de datos Cloud Firestore

La estructura de este tipo de base de datos nos permite almacenar los datos de forma jerarquizada, es decir un documento de **una colección puede contener a su vez un conjunto de colecciones**. Además de esto, como vimos en el *apartado 4.7.1* una de las características descritas y común en los sistemas *NoSQL* es la **ausencia de esquemas fijos**, es decir una misma colección puede contener documentos con diferentes esquemas, lo que ofrece una gran flexibilidad. Esta ausencia de esquemas fijos no significa que no existan ya que los datos siempre son conformes a un esquema sólo que en este caso está implícito en ellos y en el código que los maneja [47].

6.7.2 Esquema de datos

Partiendo del modelo de clases definido en la *Ilustración 6-10* y considerando las características de una base de datos *NoSql* se pretende generar un modelo de almacenamiento que de soporte a los datos. En la actualidad, y en la medida que se conoce, solo la herramienta *ER/Studio*⁶³ ofrece una visualización de esquemas *NoSQL*, y solo para el sistema *MongoDB*⁶⁴ [47]. Una representación mediante un diagrama de entidad relación⁶⁵ no representaría la estructuración real de los datos en el sistema de almacenamiento *NoSQL* de la

⁶³ **ER/Studio**: es un software de arquitectura de datos y diseño de bases de datos, es utilizado por arquitectos de datos para crear y administrar diseños de bases de datos, documentar y reutilizar activos de datos.

⁶⁴ **MongoDB**: es un sistema de base de datos *NoSQL* orientado a documentos, desarrollado bajo el concepto de código abierto.

⁶⁵ **Diagrama entidad relación**: es una representación para el modelo de datos, la cual permite representar entidades de una base de datos

forma jerarquizada que este utiliza. Es por esto, para ofrecer en este documento una representación que aporte de una visión general de la estructuración final del modelo de datos, se ha optado por utilizar la representación que podemos visualizar en la *Ilustración 6-12*.

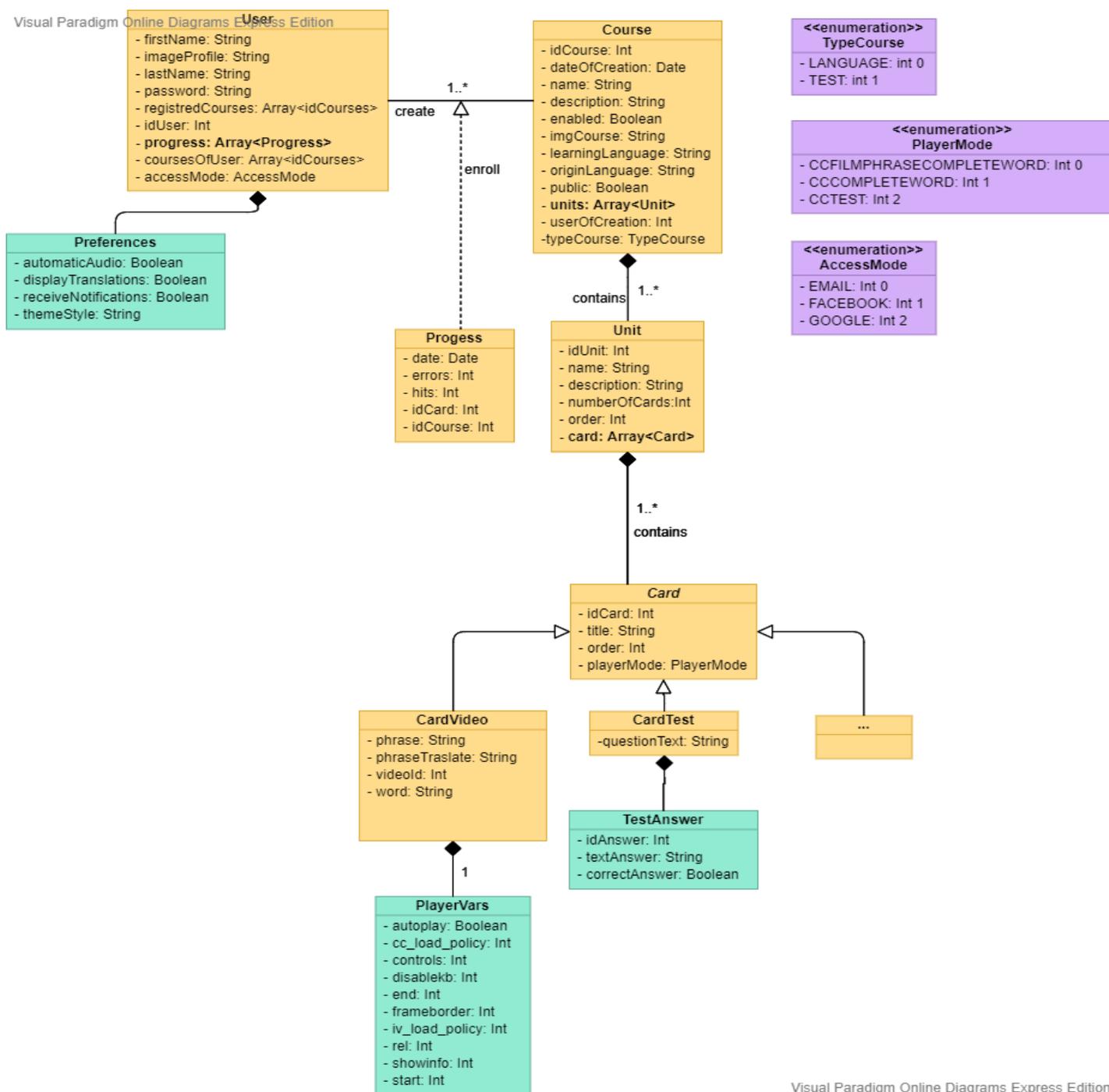


Ilustración 6-12: Diagrama de estructuración de almacenamiento de datos

Es necesario destacar que la representación que visualizamos en la *Ilustración 6-12* utiliza parte de la notación de un diagrama de clases, pero en algunos aspectos puede no ser una representación formal de UML de este tipo de diagramas. Se ha utilizado esta notación ya que es la forma más visual de representar como se estructuran de forma jerárquica los datos en nuestra base de datos *NoSQL*.

Para representar la estructuración de los datos en forma de jerarquía, es decir colecciones que a su vez contienen otras colecciones se ha optado por representarlo en la *Ilustración 6-12* aprovechando la notación de relaciones de composición y destacándolo en negrita como una propiedad de tipo Array las colecciones que contienen a su vez otro conjunto de colecciones.

Por último, hay que indicar que para obtener el total del esquema se ha utilizado un proceso de **ingeniería inversa**⁶⁶ obtenido parte del esquema de datos mediante la herramienta *JSONDiscoverer* la cual permite obtener a partir de un documento JSON, su estructura.

6.7.3 Colecciones de datos

En este apartado detallaremos la estructuración jerárquica de las principales colecciones, documentos y datos, efectuada para el almacenamiento de la información.

6.7.3.1 Colección Users

Como del propio nombre podemos deducir, en esta colección se almacenan los datos referentes a la información de cada uno de los usuarios de la aplicación. Los datos de cada usuario estarán contenidos en un único documento, de forma individualizada. Dependiendo del modo de registro en la aplicación, existirán diferentes estructuras de datos, ya que, por ejemplo, si el usuario, crea su cuenta mediante el registro de Facebook, obtendremos entre otros datos su imagen pública de perfil. Por el contrario, si el registro a la aplicación se realiza utilizando email y contraseña, no será requerida una imagen al usuario y por tanto, en el esquema del documento no existirá el campo *"imageProfile"*. Podemos ver un ejemplo de la estructuración jerárquica de esta colección en la *Tabla 6-2* donde se muestran dos tipos de documentos con la estructuración descrita.

⁶⁶ **Ingeniería Inversa:** es el proceso llevado a cabo con el objetivo de obtener información o un diseño a partir de un producto, con el fin de determinar cuáles son sus componentes y de qué manera interactúan entre sí y cuál fue el proceso de fabricación



Tabla 6-2: Estructuración Jerárquica Colección Users

6.7.3.2 Colección Courses

En esta colección se almacenarán los diferentes cursos del sistema. Podemos ver un ejemplo de la estructuración jerárquica de esta colección en la *Tabla 6-3*.

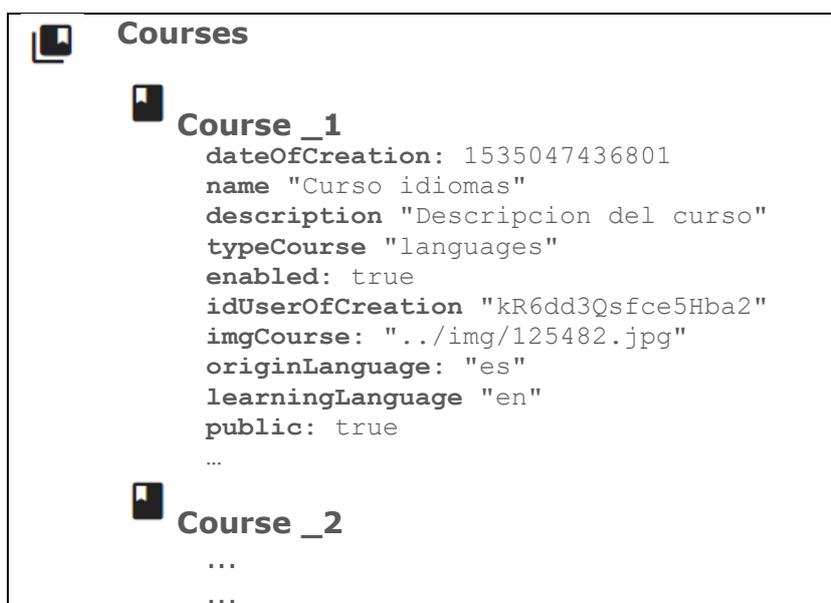


Tabla 6-3: Estructuración Jerárquica Colección Courses

6.7.3.3 Colección Units

Esta colección contiene las unidades de los diferentes cursos, estructura del documento. Esta colección se encuentra **anidada jerárquicamente** dentro de cada documento de la colección *courses*. Podemos ver un ejemplo de la estructuración jerárquica de esta colección en la *Tabla 6-4*.

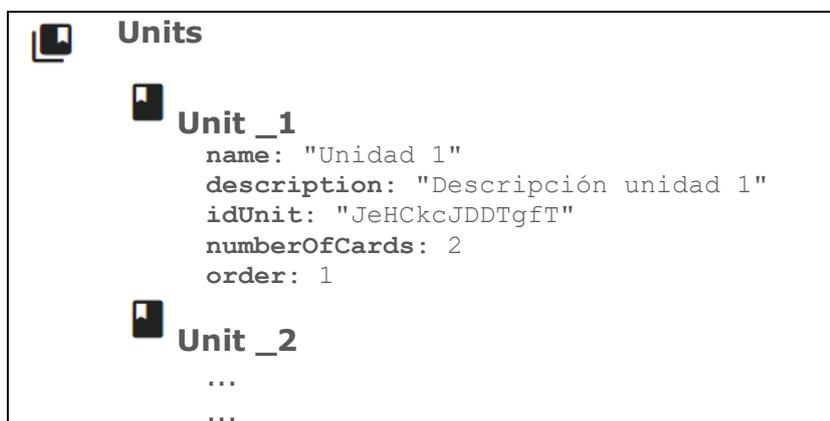


Tabla 6-4: Estructuración Jerárquica Colección Units

6.7.3.4 Colección Cards

En la colección *cards* se almacenan las diferentes tarjetas de aprendizaje de cada unidad de un curso. Esta colección se encuentra **anidada jerárquicamente** dentro de cada documento de la colección *units*. La estructura del documento varía en función del campo *mode* y que define el tipo de tarjeta de aprendizaje. Podemos ver un ejemplo de la estructuración jerárquica de esta colección en la *Tabla 6-5*.



Tabla 6-5: Estructuración Jerárquica Colección Cards

6.7.4 Seguridad de los datos

6.7.4.1 Reglas de seguridad de datos

Cloud Firestore permite definir **reglas de seguridad**, estas reglas son establecidas en el servidor de datos, por lo que son inaccesibles por parte de las aplicaciones cliente y, por tanto, aportan un **mayor nivel de seguridad**. Estas reglas de seguridad permiten el control de acceso a los datos y la validación de los datos, es decir, quien puede acceder a los datos y que tipo de datos y formato son requeridos.

Cada una de las solicitudes enviadas a la base de datos desde las diferentes aplicaciones cliente (móvil, escritorio o web) se compara con las reglas de seguridad antes de leer o escribir datos. En el caso de que alguna de las reglas definidas no fuese satisfecha, falla la solicitud completa.

Las reglas se encuentran definidas mediante declaraciones *match* que especifican la ruta del documento y una expresión *allow* que define cuando se permite la creación, la actualización, la lectura o escritura en el documento.

Como resumen de las reglas definidas en la aplicación se muestra la *Tabla 6-6* donde se exponen el conjunto total de reglas definidas. Con el objetivo de comprender su funcionamiento y el alcance de estas, son descritas dos de ellas RSD-001 y RSD-008. Por seguridad, la sintaxis del resto de las reglas de la aplicación no será expuesta en el documento ya que su publicación limitaría la continuidad del proyecto.

Código	Datos regla de seguridad
RSD-001	Descripción: Denegar el acceso de lectura/escritura a todos los documentos a cualquier usuario que no haya iniciado sesión en la aplicación.
	Colecciones afectadas: <i>Users, Courses, Units, Cards</i>
RSD-002	Descripción: Denegar el acceso de lectura/escritura a todos los documentos de la colección <i>users</i> cuyo usuario sea distinto del usuario que realiza la petición.
	Colecciones afectadas: <i>Users</i>
RSD-003	Descripción: Denegar el acceso de escritura en todos los documentos de la colección <i>Courses</i> cuyo usuario sea distinto del creador del curso.
	Colecciones afectadas: <i>Courses</i>
RSD-004	Descripción: Denegar el acceso de escritura en todos los documentos de la colección <i>Units</i> cuyo usuario sea distinto del creador de la unidad.
	Colecciones afectadas: <i>Units</i>
RSD-005	Descripción: Denegar el acceso de escritura en todos los documentos de la colección <i>Cards</i> cuyo usuario sea distinto del creador de la tarjeta de aprendizaje.
	Colecciones afectadas: <i>Cards</i>
RSD-006	Descripción: Denegar el acceso de escritura y lectura en todos los documentos de la colección <i>Courses</i> cuya propiedad <i>public</i> sea igual a <i>false</i> . Esta regla no se aplica si el usuario es el creador del curso.
	Colecciones afectadas: <i>Courses</i>
RSD-007	Descripción: Denegar el acceso de escritura y lectura en todos los documentos de la colección <i>Courses</i> cuya propiedad <i>enabled</i> sea igual a <i>false</i> . Esta regla no se aplica si el usuario es el creador del curso.
	Colecciones afectadas: <i>Courses</i>
RSD-008	Descripción: <i>Denegar el acceso creación de documentos de la colección Courses cuya validación de datos no sea superada.</i>
	Colecciones afectadas: <i>Courses</i>

Tabla 6-6: Reglas de seguridad de datos

RSD-001

Es la regla de seguridad de datos más restrictiva de la aplicación, su objetivo es denegar el acceso de lectura y escritura de usuarios no autenticados. Su sintaxis es simple y es la indicada en la *Tabla 6-7*.

```
match /{document=**} {
  allow read, write: if request.auth.uid != null;
}
```

Tabla 6-7: Regla de seguridad de datos RSD-001

RSD-008

Esta regla de seguridad de datos, valida la estructura y el contenido de los datos de un nuevo curso. Como vimos en la *sección 6.7.3.2* de este documento, esta colección tiene diferentes estructuras de datos, dependiendo del tipo de curso, pero parte de la estructura es común, por lo que en esta regla se valida que los datos comunes sean existan y sean válidos. Su sintaxis es la indicada en la *Tabla 6-8*.

```
match /{document=**} {
  allow create : if isValidCourse();
}

function isValidCourse() {
  return
  request.resource.data.dateOfCreation: != "" &&
  request.resource.data.name.size() < 50 &&
  request.resource.data.description.size() < 300 &&
  request.resource.data.type == "languages" ||
  request.resource.data.type == "test" &&
  request.resource.data.description.size() < 300 &&
  ...
  ...
}
```

Tabla 6-8: Regla de seguridad de datos RSD-008

6.8 Convención de nombres y estructura de directorios

Tras realizar algunas pruebas con el *framework* *Quasar* y el lenguaje *Vue* y antes de comenzar con el proyecto, investigué sobre cuál es la estructura de carpetas, convención de nombres para los componentes. Para esto revisé la documentación oficial de *Vue* [44], artículos [48], libros [46] [49] [26] y proyectos de código abierto de *GitHub*, con la información recabada, estructuré el proyecto en los archivos y directorios que podemos apreciar en la ilustración *Tabla 4-9*.

src/	
components/	# Componentes vue. Usados en páginas y layout.
css/	# CSS globales de la App.
layouts/	# Diseños de la App.
pages/	# Páginas de la App.
plugins/	# Plugin utilizados en la App.
router/	
index.js	# Definición de Vue Router.
routes.js	# Definición de las rutas de la App.
auth-guard.js	# Seguridad rutas de la App.
store/	
index.js	# Definición de Store Vuex.
courses	# Vuex Store Modulo Cursos.
units	# Vuex Store Modulo Unidades.
users	# Vuex Store Modulo Usuarios.
cards	# Vuex Store Modulo Tarjetas de aprendizaje.
App.vue	# Componente root de la App (Configuración DB)
cloud-functions	# Funciones Firebase.
test-selenium	# Pruebas selenium.
firebase.rules	# Reglas seguridad Firestore DB.
src-pwa/	# Código específico PWA.
src-cordova/	# Código para generar Apps móviles.
src-electron/	# Código para App escritorio.
dist/	
pwa-mat/	# Distribución App generada PWA.
electron-mat/	# Distribución App generada Escritorio.
quasar.conf.js	# Configuración framework Quasar.
.gitignore	# Especifica los archivos Git ignorados.
package.json	# npm scripts y dependencias del proyecto.
README.md	# Información del proyecto y archivos.

Tabla 6-9: Estructuración de los archivos del proyecto

Referente con la **convención de nombres** utilizada para denominar los archivos del proyecto, se ha optado por una notación *UpperCamelCase* en la que, los nombres de los ficheros se componen de una o varias palabras identificativas, en la que se establece a modo de separador la primera letra

de cada palabra en mayúscula. Además, para los componentes (*components*) que conforman la aplicación se ha optado por anteponer las siglas “Cc” al nombre del componente, diferenciando de esta forma los componentes de los diseños (*layouts*) y paginas (*pages*). Podemos ver un ejemplo de esto en la *Tabla 6-10*.

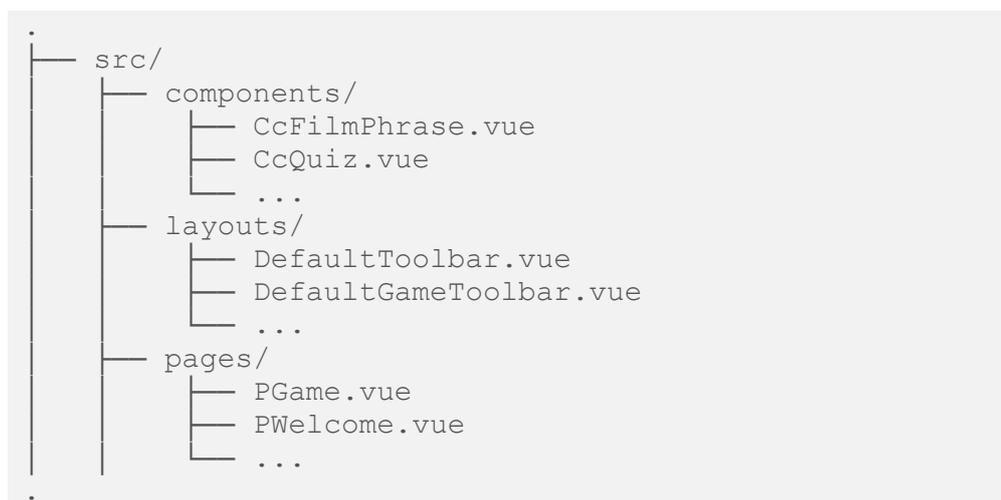


Tabla 6-10: Convención de nombres

6.9 Tipos de usuarios en el sistema

Los usuarios de la aplicación pueden realizar diferentes tareas en función del tipo de usuario bajo el que estén definidos. Cada uno de los usuarios de la aplicación se encuentran definidos dentro de un grupo de usuarios, los cuales tienen el derecho o no al uso de ciertas características. Encontramos dos tipos de usuarios que definimos a continuación.

Participante: Usuario con capacidad para crear y modificar los cursos creados por el mismo. Este usuario tiene acceso para visualizar y matricularse en los cursos creados por el resto de los usuarios.

Super Usuario: Usuario con capacidad para crear y modificar los cursos creados por el mismo y otros usuarios. Este tipo de usuario estará a cargo de moderar y supervisar el contenido de los cursos creados en la aplicación.

En la *Tabla 6-11* se definen las funcionalidades que cada modalidad de usuario puede efectuar.

	Participante	Super Usuario
Crear nuevos cursos.	SI	SI
Modificar cursos propios.	SI	SI
Modificar cursos ajenos.	NO	SI
Editar información de otros usuarios.	NO	NO
Editar información de su perfil.	SI	SI
Matricularse en cursos propios.	SI	SI
Matricularse en cursos públicos de otros usuarios.	SI	SI
Modificar unidades de cursos propios.	SI	SI
Modificar unidades de cursos ajenos.	NO	SI
Añadir tarjetas de aprendizaje en cursos propios.	SI	SI
Añadir tarjetas de aprendizaje en cursos ajenos.	NO	SI

Tabla 6-11: Tipos de usuarios y permisos

6.10 Prototipo Inicial

En el siguiente prototipo inicial, se pretende dar una visión clara de las principales funcionalidades de la aplicación, todas las interfaces constan de dos diseños, un primer diseño para dispositivos de pantalla reducida (resolución inferior 768px), y un segundo diseño para dispositivos de una resolución superior. Mediante estos dos diseños se pretende cumplir con el requisito no funcional **RNF-003 - Interfaces de usuario** descrito en el apartado 5.4.3 de este documento.

Resaltar que la funcionalidad en ambos diseños es equivalente, pero adaptada mediante un diseño responsivo al tipo de pantalla, intentando con esto otorgar siempre al usuario el **mayor grado de usabilidad** posible de las interfaces diseñadas.

6.10.1 Pantalla inicial

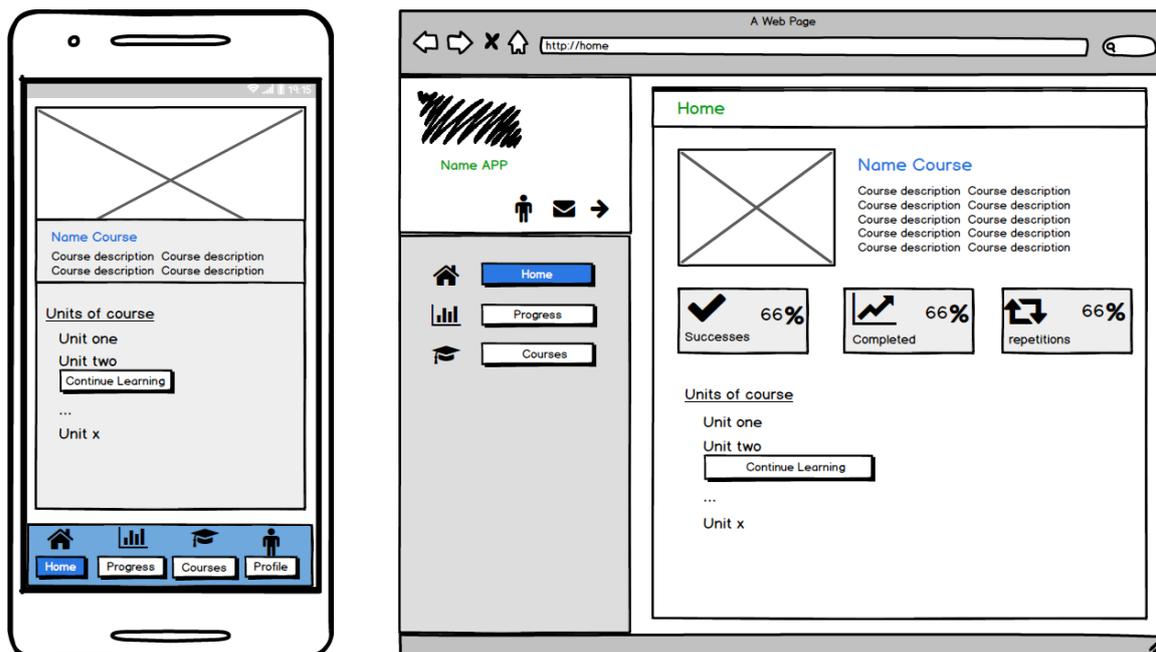


Ilustración 6-13: Prototipo Inicial - Pantalla Inicial

Es la vista disponible tras realizar la autenticación o registro en la aplicación. En ella aparecerá un resumen claro del avance del curso actual en el que el usuario este matriculado, dando la posibilidad de continuar el aprendizaje por la unidad en la que finalizo su última sesión de estudio.

6.10.2 Pantalla de progreso

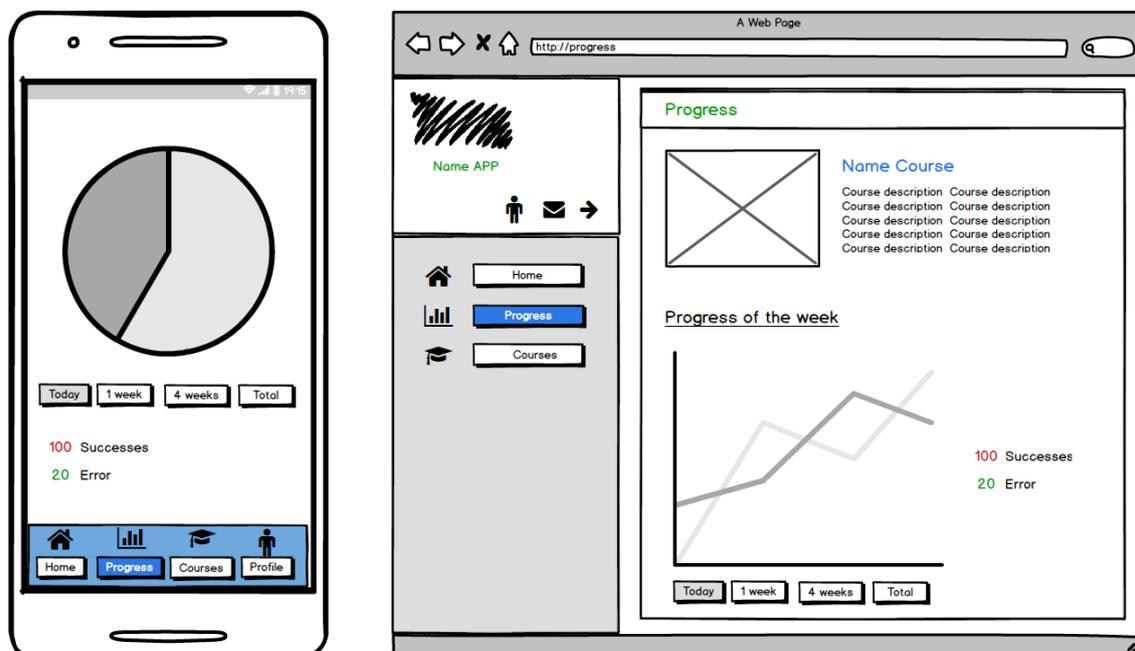


Ilustración 6-14: Prototipo Inicial Pantalla de progreso

En esta vista de la aplicación se mostrarán los datos referentes al progreso dentro del curso, esta información será mostrada mediante gráficos, que podrán mostrar datos referentes al progreso del día, la semana, el mes o el total del curso.

6.10.3 Pantalla de cursos

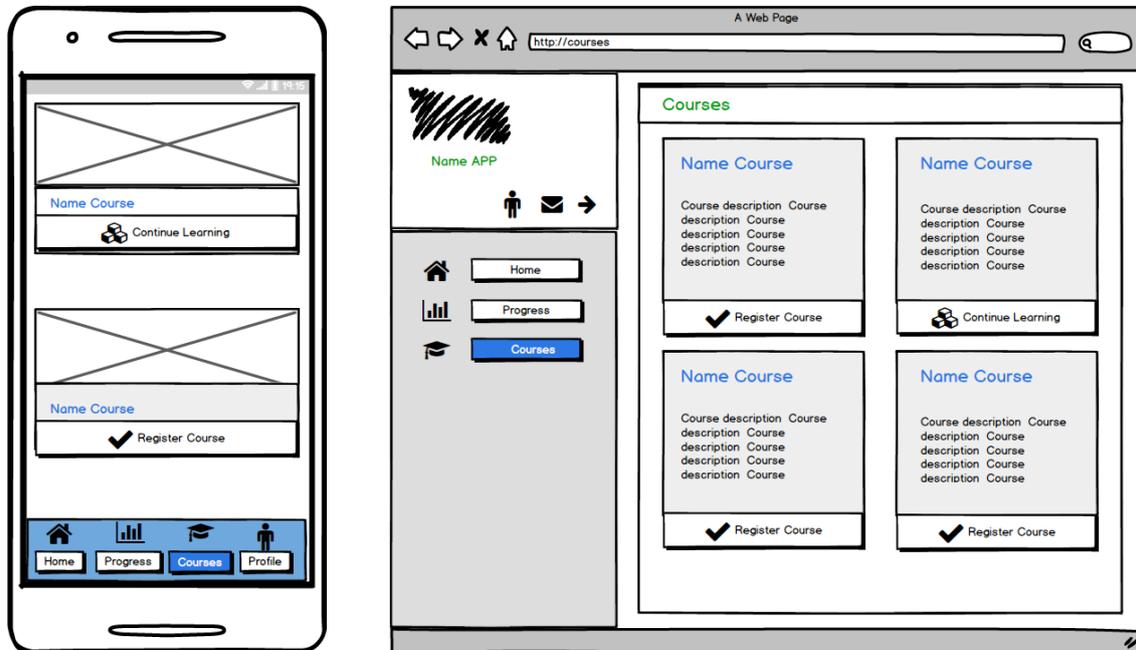


Ilustración 6-15: Prototipo Inicial - Pantalla Cursos

En esta vista de la aplicación estarán disponibles los cursos en los que el usuario este matriculado, se ofrecerá la posibilidad de cambiar entre los cursos matriculados, además se mostrarán nuevos cursos ordenados por popularidad en los que el usuario no está matriculado. Desde esta vista se ofrecerá la posibilidad de crear nuevos cursos.

6.10.4 Pantalla de tarjeta de aprendizaje

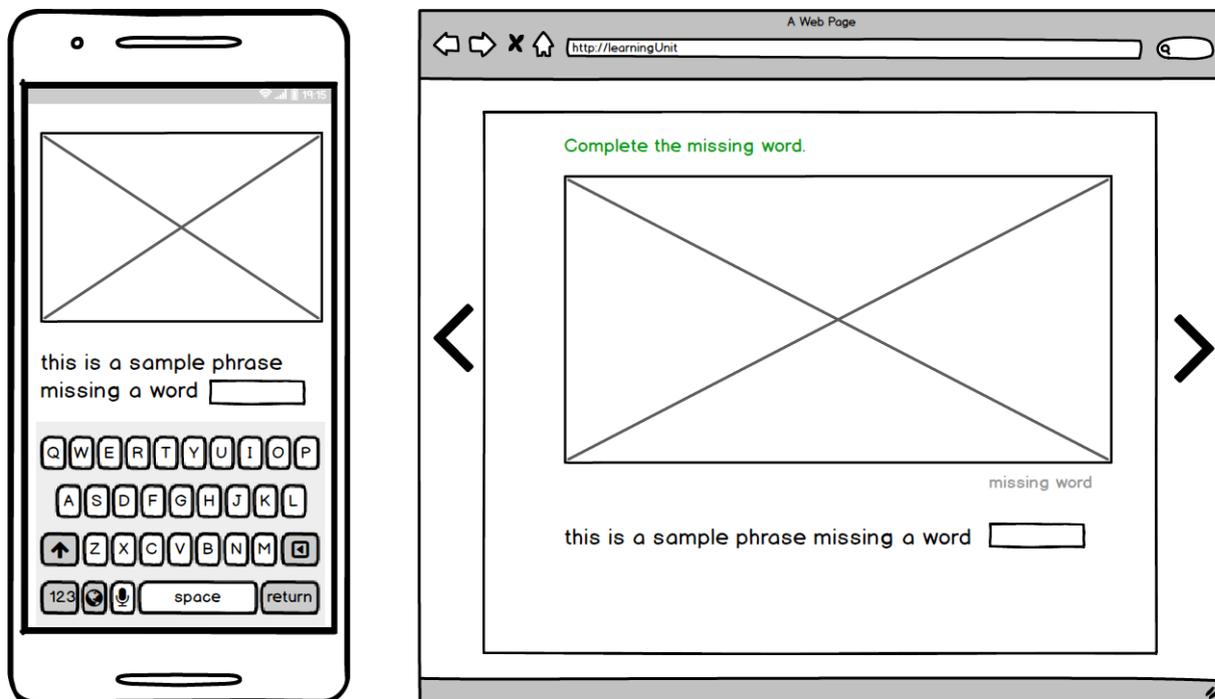


Ilustración 6-16: Prototipo Inicial - Pantalla de tarjeta aprendizaje

Esta vista contiene las tarjetas de aprendizaje que forman una unidad de un determinado curso, además se incluirán las tarjetas pendientes de repasar por parte del usuario. Existirán diferentes tarjetas de aprendizaje por lo que los diseños pueden variar en función del contenido de cada modalidad de tarjeta.

6.11 Paleta de colores y tipología de letra empleada

La paleta de colores y tipografías de la aplicación han sido escogidas con el objetivo **de facilitar la usabilidad y legibilidad** de la aplicación haciendo que la navegación por la aplicación sea la apropiada para el usuario.

Para la elaboración del t emplate se ha intentado ajustarse al uso de las siguientes paletas de colores:



Paleta de colores Primaria con c digo color hexadecimal



Paleta de colores secundaria con c digo de color hexadecimal

En cuanto a la paleta de colores primaria ha sido empleada para el dise o general del sistema. La paleta de colores secundaria ha sido utilizada en las notificaciones y botones en los que se precisa llamar la atenci n del usuario.

Bien es cierto que en algunos casos no se ha podido respetar en su totalidad el uso de las paletas de colores seleccionadas en un inicio ya que, al tratarse en su mayor a de una paleta de colores claros, existen monitores en los que los colores var an y tienden a verse casi blancos por lo que se opt  por realizar algunas modificaciones.

En referencia con la tipolog a de letra empleada, en su mayor a es *sans-serif* ya que se trata de una tipolog a de letra "sin remates" esto permite que en la pantalla la pixelaci n se vea de forma mucho m s limpia que otro tipo de tipolog a con remates.

6.12 Implementación del Algoritmo de repetición espaciada

6.12.1 Introducción a la planificación de la repetición

El objetivo de una planificación de repeticiones es conseguir determinar de forma ajena a la intervención por parte del usuario, cual es el **momento óptimo para volver a repasar** una tarjeta de aprendizaje. En nuestra aplicación se dispone de diferentes cursos cuyo contenido está conformado por unidades, cada una de estas unidades dispone de un conjunto de tarjetas de aprendizaje, siendo estas la unidad mínima de aprendizaje posible y sobre la que efectuaremos la planificación.

Esta planificación tiene lugar por primera vez, cuando un usuario accede a un curso y realiza las tarjetas de aprendizaje contenidas en una unidad del curso, para cada una de ellas se va realizando una **planificación de repetición**. Esta planificación es determinada en función de la calidad de la respuesta. La próxima vez que el usuario vuelva a utilizar el sistema, se realizara una comprobación sobre que tarjetas de aprendizaje están pendientes de ser repasadas dependiendo de la fecha para la que estuvieran planificadas respecto a la fecha actual. Estas tarjetas repasadas volverán a ser replanificadas para una nueva fecha teniendo en cuenta los mismos criterios.

Para realizar esta planificación de forma automática en nuestra aplicación, se ha optado por utilizar una adaptación de la versión dos del algoritmo definido por *Piotr Woźniak*, esta versión del algoritmo es denominado **SM-2** y lo definen como un método para calcular el momento óptimo de repetición del contenido aprendido. Las principales ventajas de este algoritmo frente a otros métodos como *Leitner* fueron descritas en el *apartado 3.2.1* por lo que en los consiguientes apartados nos centraremos en dar una visión global del algoritmo SM-2 y las modificaciones efectuadas para dar cabida al algoritmo en nuestro sistema.

6.12.2 El algoritmo SM-2 genérico

El proceso de programación efectuado por el algoritmo SM-2 comienza después de aportar la respuesta en cada tarjeta de aprendizaje, el algoritmo utiliza una escala entre 0 y 5 para determinar la "calidad de respuesta" siendo 0 o 1 si el usuario no sabe la respuesta o lo ha olvidado por completo. El algoritmo continuará repitiendo la tarjeta hasta que el usuario obtenga una calificación de 2 o superior, considerando en este momento el algoritmo que el usuario es capaz de recordar el contenido de la tarjeta como mínimo un

día, siendo programada una repetición en tal fecha. Este punto señala la **transición de la memoria de corto⁶⁷ a largo plazo⁶⁸**.

Para las tarjetas de nuevo aprendizaje, es decir las tarjetas que son vistas por primera vez, se establece una primera repetición que tendrá lugar siempre al día siguiente y otra transcurridos 6 días. Esto solventa el conocido como "problema de arranque en frío" [50].

Transcurrida la fecha de programación de la repetición, la tarjeta será mostrada nuevamente al usuario para su repaso, determinándose nuevamente la calidad de la respuesta: si la calificación es superior a 3 se programará una nueva repetición aumentándose el intervalo. Por el contrario, si la calificación es inferior, este intervalo disminuirá programándose repeticiones a corto plazo.

Una representación de esta metodología de programación de repeticiones empleada por el algoritmo SM-2 podemos verla en la *Ilustración 6-17*.

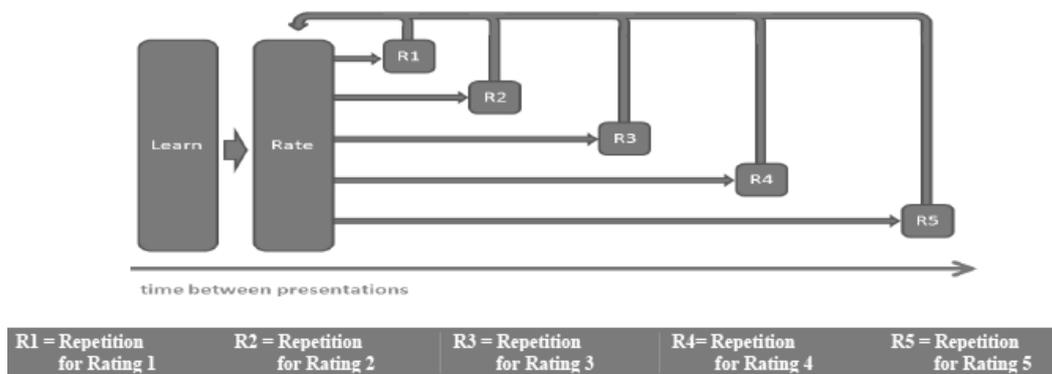


Ilustración 6-17: Prolongación de los intervalos de repetición [50]

Esta metodología en la programación de las repeticiones se basa en la aproximación matemática conocida como **curva del olvido** y que vimos en la *sección 3.2*, la cual defiende que los conceptos más arraigados en la memoria pueden tener un tiempo más largo de repetición.

Una aproximación de los pasos del algoritmo SM-2 se detallan en la *Tabla 6-12*.

⁶⁷ **Memoria a corto plazo:** es la capacidad para mantener en mente de forma activa una pequeña cantidad de información, de forma que se encuentre inmediatamente disponible durante un corto periodo de tiempo

⁶⁸ **Memoria a largo plazo:** es la capacidad de almacenar recuerdos por un plazo de tiempo mayor a seis meses, sin que se le presuponga límite alguno de capacidad o duración

Definición de la acción	
1	Divida el conocimiento en los elementos más pequeños posibles.
2	EF:= 2.5 para todas las tarjetas de aprendizaje
3	Las tarjetas serán repetidas siguiendo los siguientes intervalos: I(1):= 1 I(2):= 6 para n>3: I(n):= I(n-1)*EF <u>donde:</u> I(n) – intervalo de repetición (en días). EF: Factor E de un elemento dado. Si el intervalo es una fracción, redondee al entero más cercano.
4	Después de cada repetición evalúe la calidad de la respuesta en una escala de grado de 0-5, donde: 5 – respuesta correcta. 4 – respuesta correcta: después de una vacilación. 3 – respuesta correcta: recordada con dificultad grave. 2 – respuesta incorrecta: donde el correcto parecía fácil de recordar 1 – respuesta incorrecta. 0 – respuesta en blanco.
5	Después de cada repetición, modifique el factor E del elemento repetido recientemente según la fórmula: EF' = EF + (0.1 - (5-q) * (0.08 + (5-q) * 0.02)) <u>donde:</u> EF' – Nuevo valor del factor E. q – Calidad de la respuesta en escala de grados 0-5. Si EF es menor que 1.3 entonces EF es 1.3.
6	Si la evaluación de calidad (q) es menor que 3, comience las repeticiones desde el principio sin cambiar EF (es decir, use los intervalos I(1) , I(2) , etc. Como si el elemento se hubiera memorizado de nuevo)
7	Después de cada sesión de repetición de un día determinado, repita nuevamente todos los ítems que obtuvieron una puntuación por debajo de cuatro en la evaluación de calidad(q). Continúe las repeticiones hasta que todos estos elementos obtengan al menos cuatro puntos.

Tabla 6-12: Secuencia de acciones algoritmo SM-2

6.12.3 Modificaciones efectuadas del algoritmo SM-2

Las variaciones en el algoritmo SM-2 son efectuadas en la mayoría de las aplicaciones que se basan en este algoritmo de repetición, una de ellas es *Anki*, una aplicación referente en repetición espaciada y de la que hablamos en el capítulo 3. *Anki* utiliza SM-2 como algoritmo base, pero con algunas modificaciones, algunas de ellas son adoptadas en nuestro sistema y son detalladas a continuación: [21]:

- Los fallos sucesivos mientras las cartas están en aprendizaje, no resultan en disminuciones adicionales de la facilidad de la tarjeta. Esto es una queja entre los usuarios de algoritmos SM estándar, ya que fallos repetidos en tarjetas provoca una planificación repetida excesiva que hacen que la tarjeta se estanque.
- SM-2 utiliza el intervalo de 0-5 para establecer la calidad de la respuesta de cada una de las tarjetas, en nuestro caso al igual que *Anki* adoptaremos solo 4 opciones con una única opción de fallo en lugar de 3.

Además de las modificaciones descritas se pretende en una versión posterior, dar la posibilidad al usuario de repasar unidades a su elección tantas veces como desee. Esto supondría un inconveniente ya que implicaría que el usuario pueda repasar contenido previo a la planificación programada por el algoritmo SM-2. Para solventar este problema se pretende implementar un algoritmo auxiliar para admitir las repeticiones fuera de la programación establecida y que estas no afecten a la programación efectuada por SM-2, este tipo de algoritmos auxiliares son comunes en juegos de aprendizaje [50].

6.12.4 Atributos para la planificación de la repetición

Factor E (EF): Es el factor de facilidad, es decir, este refleja la facilidad de retener una tarjeta de aprendizaje en la memoria. Inicialmente todas las tarjetas de aprendizaje de la aplicación tienen un valor por omisión de 2.5 determinando este un nivel de dificultad "fácil". Este valor se va actualizando en función de la fórmula:

$$EF' = EF + (0.1 - (4 - q)) * (0.08 + (4 - q) * 0.02)$$

Este atributo tiene como valor mínimo 1.3 ya que como explica el autor del algoritmo [12], valores inferiores tienden a programar repeticiones poco efectivas para la memorización a largo plazo y que acaban siendo molestas para el usuario.

Calidad de la respuesta (q): este atributo tiene como objetivo reflejar el grado de acierto de respuesta del usuario en una determinada tarjeta de aprendizaje. Los valores oscilan en un rango de [0,3] donde:

- 0 – respuesta en blanco.
- 1 – respuesta incorrecta.
- 2 – respuesta correcta recordada con dificultad grave.
- 3 – respuesta correcta.

6.12.5 Flujograma de planificación de repeticiones

Por ultimo y con el objetivo de dar una visión más clara sobre el comportamiento del algoritmo en nuestro sistema, se incluye la *Ilustración 6-18*, donde se representa el comportamiento del algoritmo en el proceso de programación de repeticiones en las tarjetas de aprendizaje. En este flujograma es denotado el inicio con un círculo negro y el fin mediante un doble círculo blanco-negro.

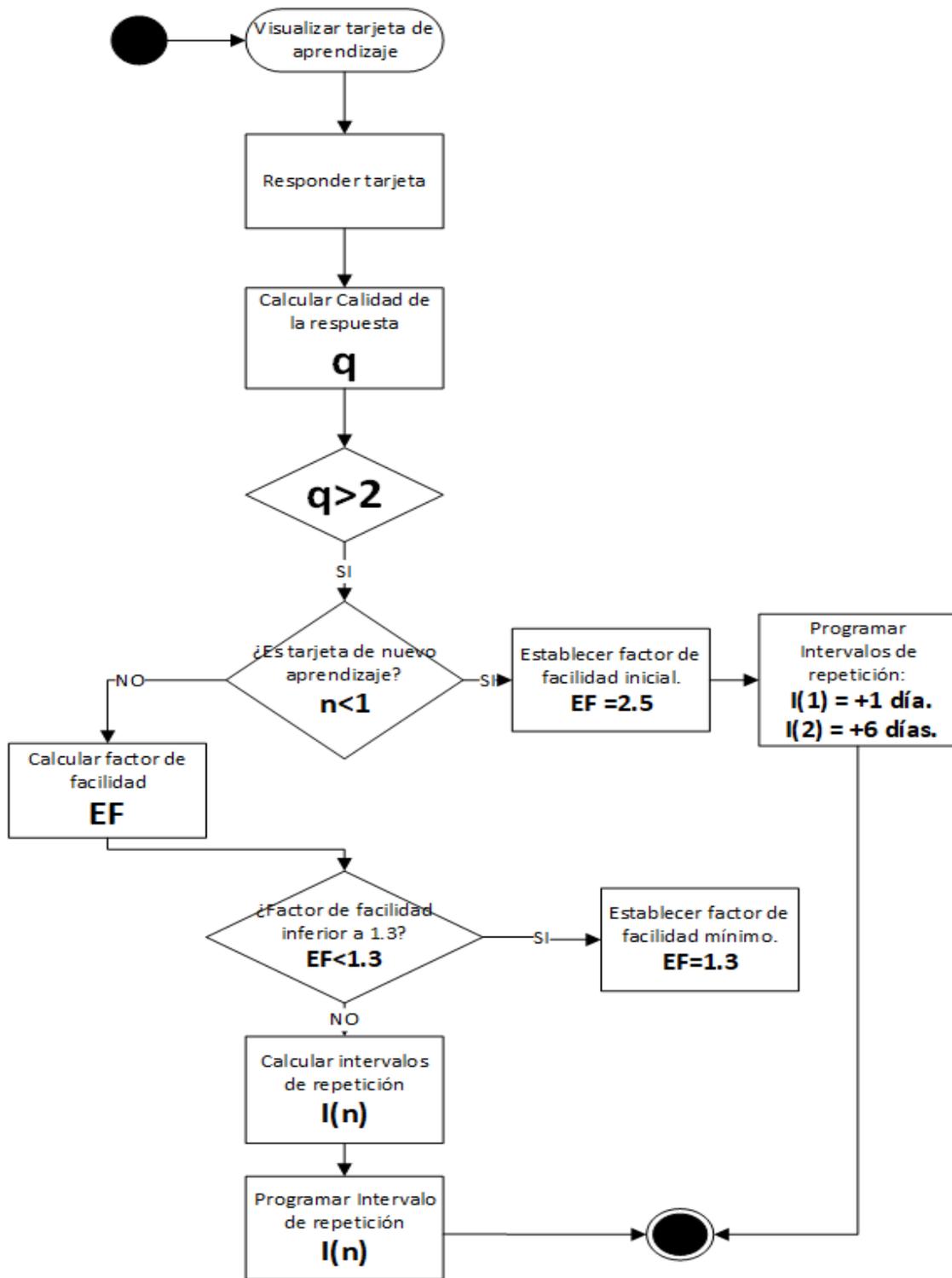


Ilustración 6-18: Flujograma programación de repeticiones

6.13 Las tarjetas de aprendizaje

En Lembra la unidad mínima de aprendizaje son las tarjetas o también conocidas como *flash cards*. Estas normalmente contienen palabras, imágenes, audio o videos y se pide al usuario que realice cierta interacción con el objetivo de memorizar su contenido. Actualmente existen dos tipos de tarjetas de aprendizaje, tarjetas de tipo test que describiremos en la *sección 6.13.1* y tarjetas de tipo video que describiremos en la *sección 6.13.2*.

Independientemente del tipo de tarjeta de aprendizaje existen elementos comunes a todas las tarjetas. Uno de estos es la barra superior y que podemos identificar en la *Ilustración 6-19* esta permite conocer el avance en el número de tarjetas completadas y restantes cuando el usuario está repasando o aprendiendo las tarjetas de una unidad de un curso. Otro de los elementos comunes es la barra inferior representada en la *Ilustración 6-19* y que posibilita el avance, retroceso y validación del contenido de las tarjetas

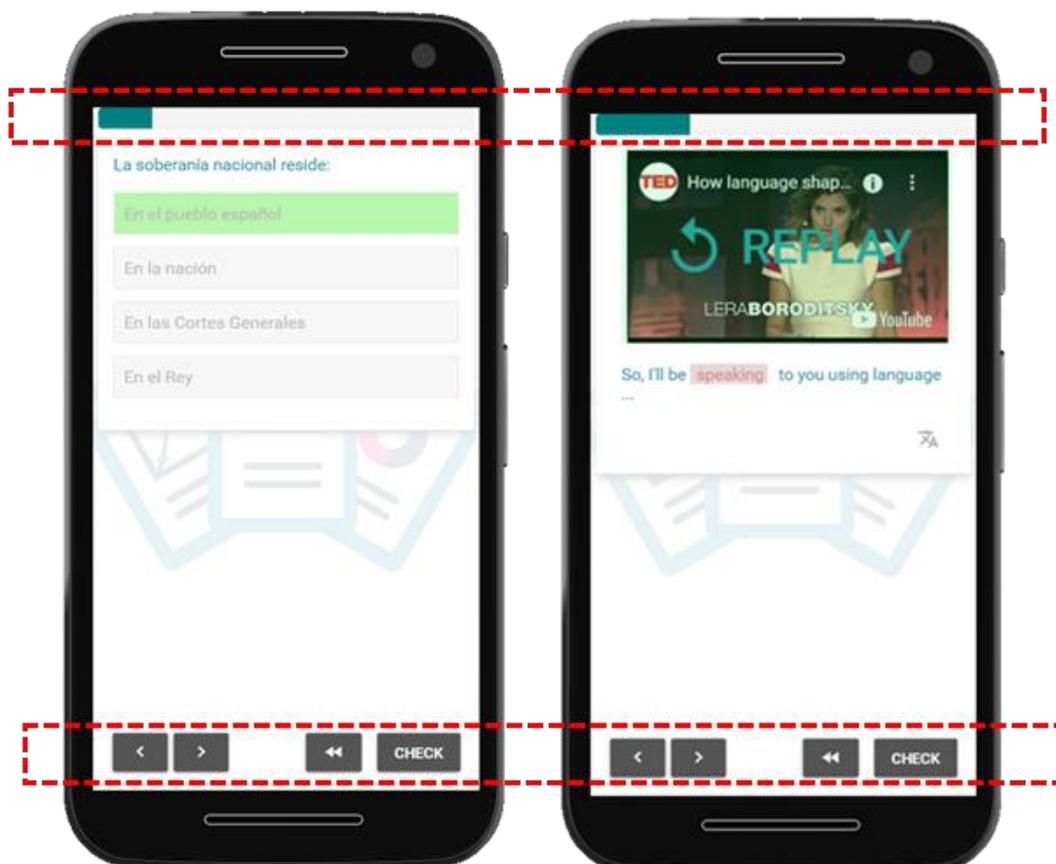


Ilustración 6-19: Contenido común de tarjetas

6.13.1 Tarjetas de tipo test

En la *Ilustración 6-20* podemos apreciar una tarjeta de tipo test compuesta por cuatro opciones y siendo solo una la correcta. El objetivo de este tipo de tarjetas es seleccionar la opción correcta de entre las mostradas. En caso de acertar la opción se marcará en color verde. Por el contrario, si la respuesta no es la correcta, esta se marcará en color rojo.

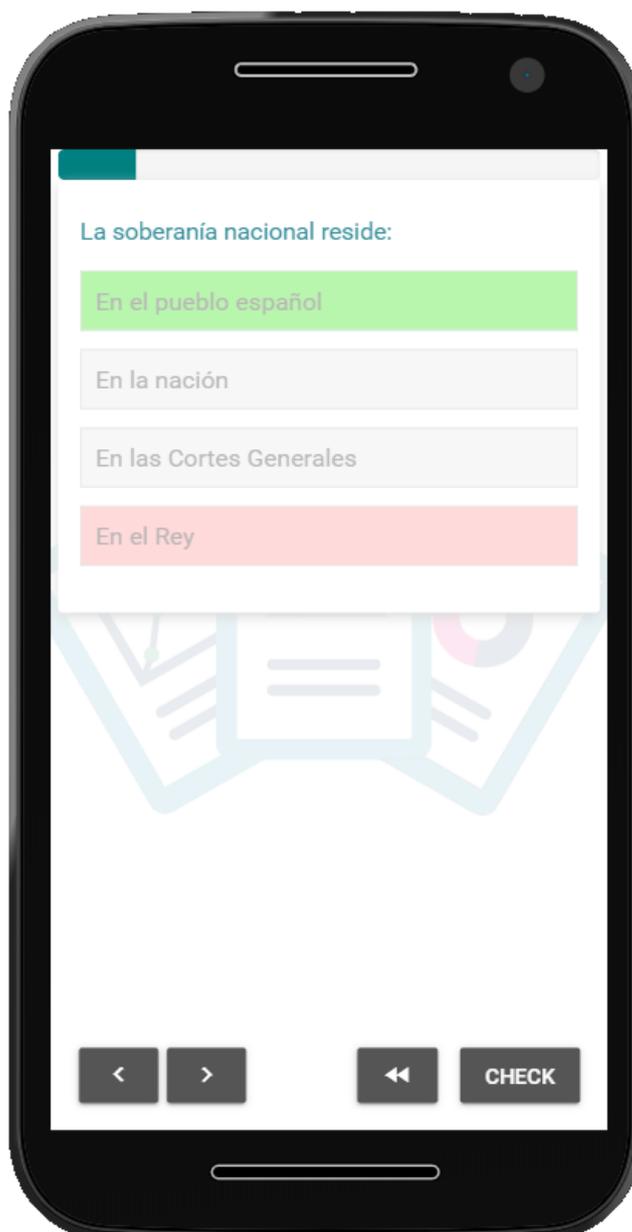


Ilustración 6-20: Tarjeta tipo test

La creación de tarjetas de tipo test se realiza desde el asistente de creación de tarjetas de tipo test. En la *Ilustración 6-21* se representa la primera ventana del asistente, cómo podemos apreciar se solicita se introduzca el texto de la pregunta.

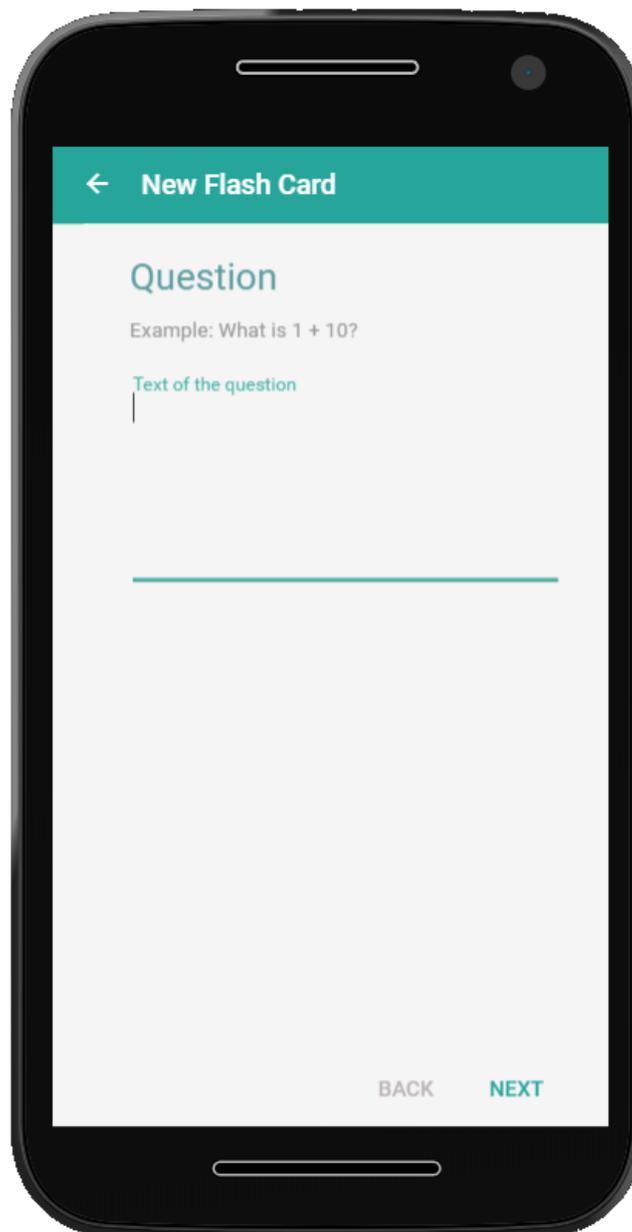


Ilustración 6-21: Asistente tarjetas tipo test.

Las siguientes ventanas del asistente nos solicitarán las diferentes opciones de respuesta a la pregunta realizada indicándose entre ellas cual es la opción correcta.

6.13.2 Tarjetas de tipo subtítulos de video

En la *Ilustración 6-22* podemos apreciar una tarjeta de tipo video, estas tarjetas reproducirán un fragmento de un video de forma automática y nos solicitara se introduzca una de las palabras ocultas del fragmento de video <5>. Mediante el botón <6> podemos ver la traducción de la palabra oculta y mediante el botón <4> podemos repetir el fragmento de video nuevamente.

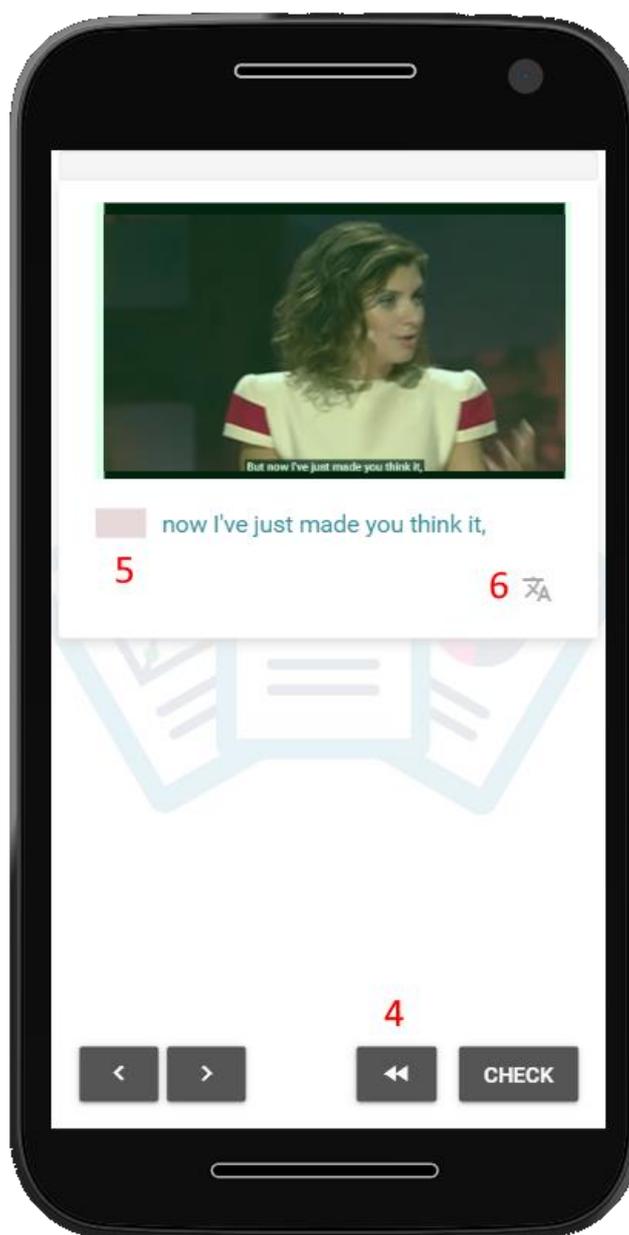


Ilustración 6-22: Tarjeta de tipo video

La creación de tarjetas de tipo video se realiza desde un asistente. En la *Ilustración 6-23* podemos ver la primera pantalla del asistente donde se nos solicita se introduzca una o varias palabras que serán utilizadas como criterio

de búsqueda en el portal de almacenamiento de contenido en video *Youtube*. Esta búsqueda de videos se realiza mediante la API de *Youtube* y de la hablamos en el *apartado 4.8.2*

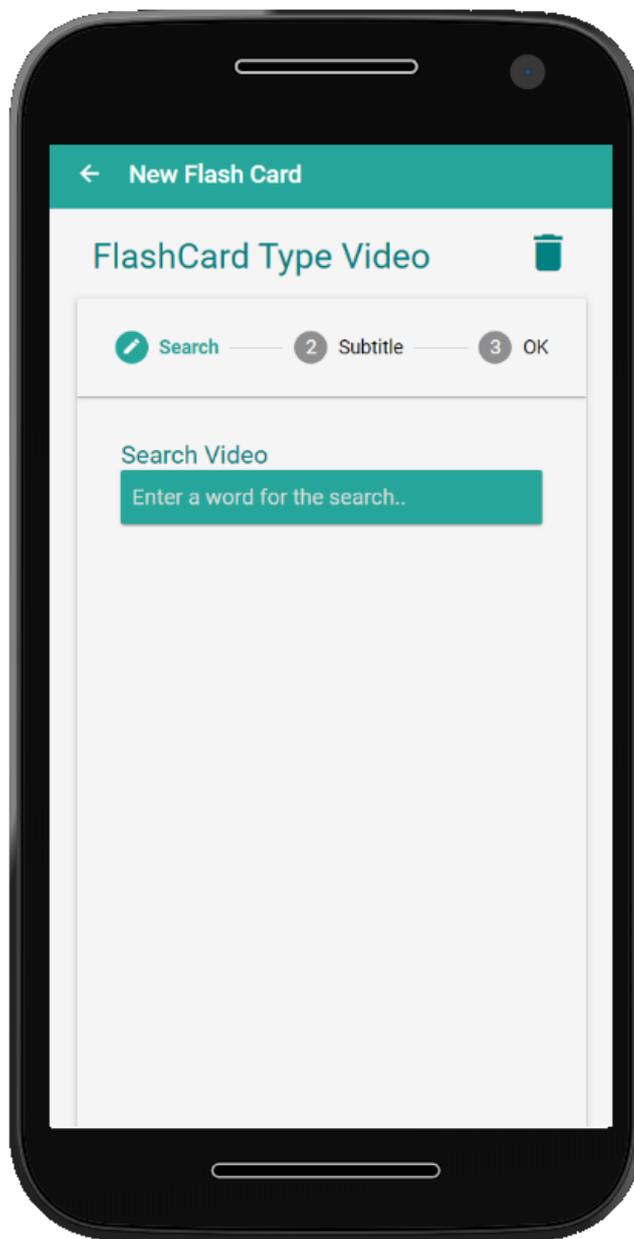


Ilustración 6-23: Asistente tarjeta video 1

Como podemos apreciar en la *Ilustración 6-24* esta búsqueda nos retornara diferentes videos como resultado de los cuales deberemos seleccionar uno de ellos.

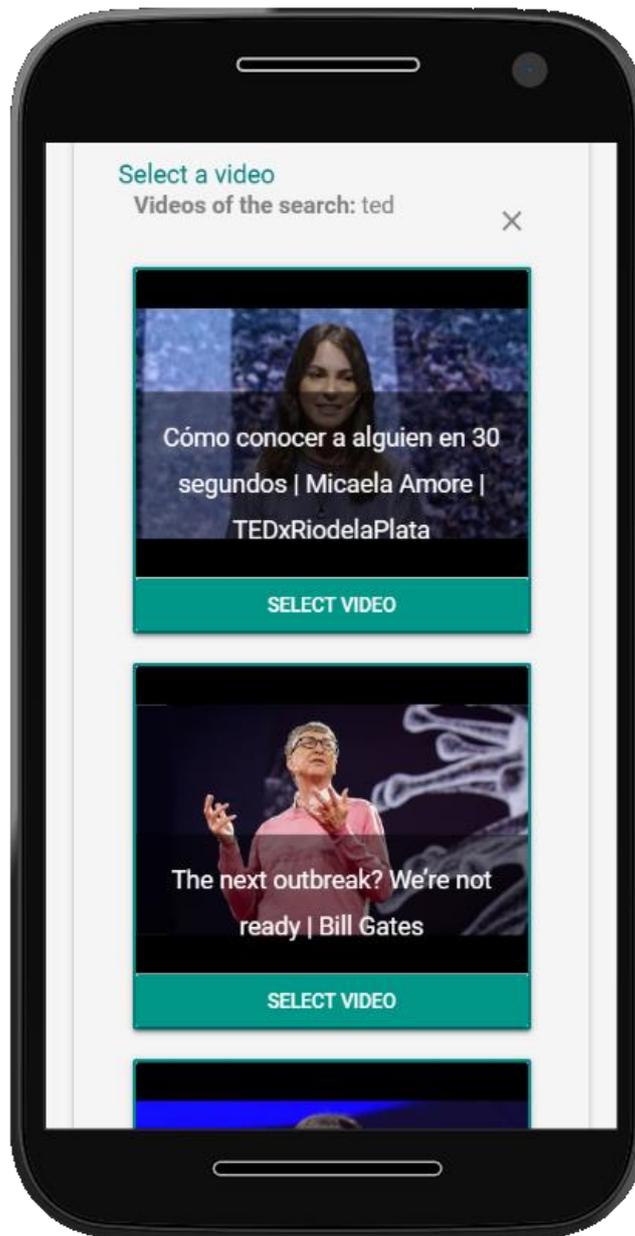


Ilustración 6-24: Asistente tarjeta video 2

Tras la selección de un video serán obtenidos en caso de existir todos los subtítulos referentes con dicho video agrupados por fragmentos. Estos son obtenidos de la API de *Youtube* la cual los proporciona en formato XML⁶⁹. Podemos ver un ejemplo en la *Tabla 6-13* en el que se aprecia como se denota el tiempo de inicio y la duración de cada uno de los fragmentos.

⁶⁹ **XML:** (Lenguaje de Marcado Extensible) es una especificación de W3C como lenguaje de marcado de propósito general. Esto significa que, a diferencia de otros lenguajes de marcado, XML no está predefinido, por lo que debes definir tus propias etiquetas. El propósito principal del lenguaje es compartir datos a través de diferentes sistemas, como Internet.

```
<transcript>
  <text start="29.411" dur="6.108">
    If I got to know at least 10 things about a person
  </text>
  <text start="35.519" dur="3.671">
    I would eat the only cream puff pastry.
  </text>
  ...
  ...
  <text start="43.289" dur="2.948">
    I set the stopwatch and I went straight to Instagram.
  </text>
</transcript>
```

Tabla 6-13: XML subtítulos API Youtube

Los fragmentos de video obtenidos de la API de *YouTube* son representados en la aplicación como podemos ver en la *Ilustración 6-25* donde el usuario debería seleccionar uno de estos fragmentos como siguiente paso del asistente de creación.

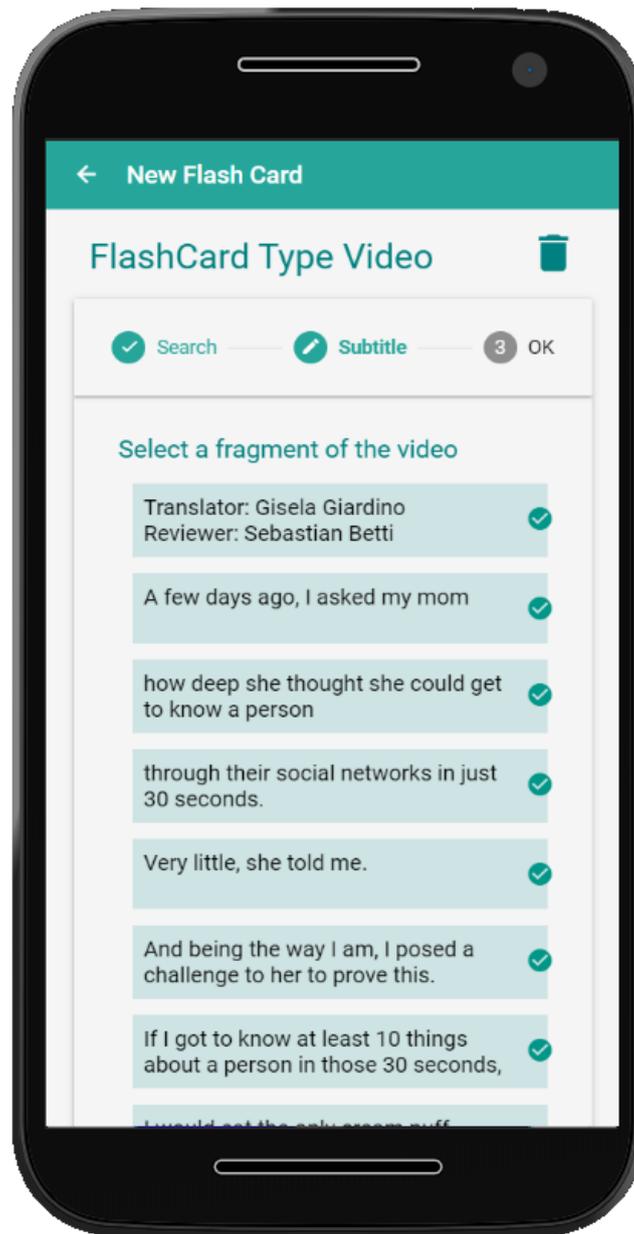


Ilustración 6-25: Asistente tarjeta video 3

Tras seleccionar un fragmento de subtítulo, el asistente de creación nos guiara al último paso del asistente y que podemos ver representado en la *Ilustración 6-26*. Esta muestra en la parte superior el fragmento de video correspondiente con el subtítulo seleccionado. Este fragmento de video es ajustable mediante el botón <8> pudiéndose ajustar el tiempo de inicio y fin del video en caso de que este no se correspondiese con el fragmento del subtítulo mostrado.

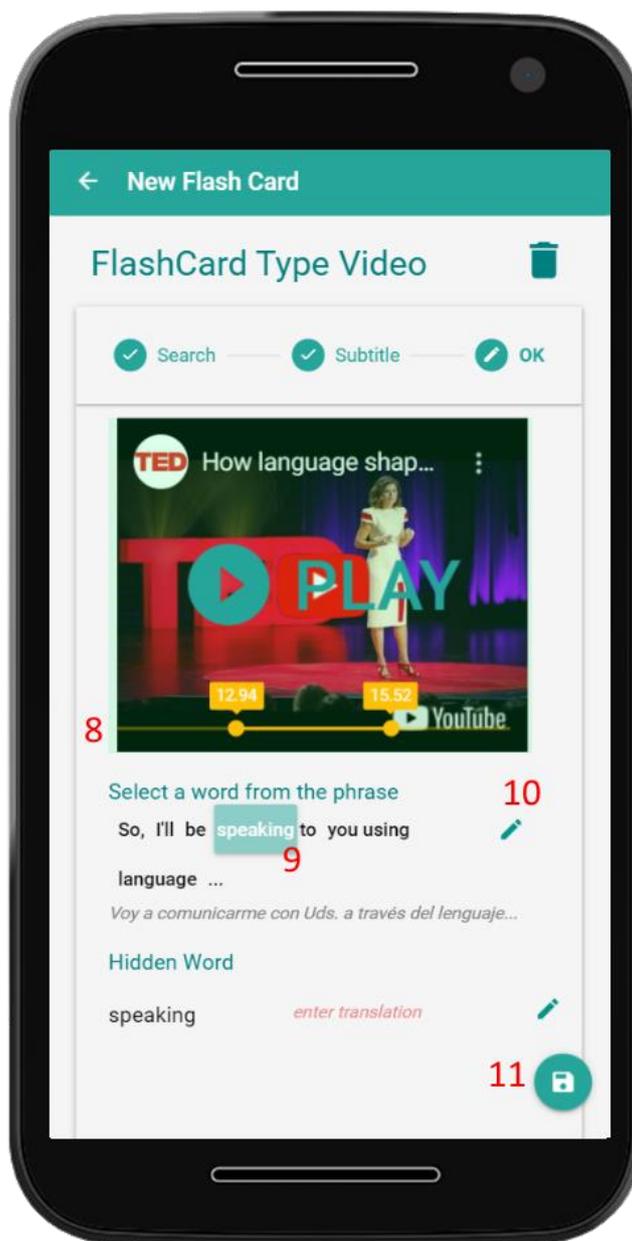


Ilustración 6-26: Asistente tarjeta video 4

En la parte inferior del video aparecerá el texto del fragmento del subtítulo y del que deberemos seleccionar una palabra <9>, esta será la palabra que el usuario con posterioridad deberá introducir atendiendo al audio del video. Para finalizar, mediante el botón <11> se guardará la tarjeta de tipo video creada.

6.13.3 Generar nuevas modalidades de tarjetas de aprendizaje

Desde un inicio el objetivo de la construcción del sistema siempre ha estado enfocado para que resulte sencillo incluir nuevas tarjetas de aprendizaje. Para

esto, se ha realizado una encapsulación del código que realiza la funcionalidad común para todas las tarjetas de aprendizaje consiguiéndose de esta forma que incluir nuevas tarjetas resulte algo sencillo y que no implique procesos complejos como pueden ser la generación de nuevos modelos de datos o estructuras que den soporte a las necesidades de nuevas tarjetas de aprendizaje.

Como vimos en la *sección 6.7.3.4* la colección *Card* que almacena el contenido de las tarjetas de aprendizaje tiene propiedades comunes para todos los tipos de tarjetas y una propiedad *data* que da libertad de almacenar una estructura que se adecua a las necesidades de cada tipo de tarjeta. Es por esto, para incluir una nueva modalidad de tarjetas de aprendizaje, bastaría con incluir dos archivos en la APP y que describimos en los apartados 6.13.3.1 y 6.13.3.2.

6.13.3.1 Archivo para el asistente de creación de la tarjeta

Este es el primer archivo que debemos incluir en la APP, y contiene el asistente o pasos que permiten crear y almacenar la tarjeta de aprendizaje en el sistema. En la *Tabla 6-14* vemos un ejemplo de cómo se realiza en la creación de una tarjeta de tipo test y que debe realizarse en la creación de cualquier tarjeta de aprendizaje únicamente modificando las partes denotadas en color azul. Estas denotaciones corresponden en primer lugar con la propiedad *mode* que identifica el tipo de tarjeta y en segundo lugar la propiedad *data* que contiene el contenido de la tarjeta. En esta propiedad *data* el creador de la tarjeta tiene la libertad de definir la estructuración jerárquica que mejor se adecue al contenido necesario de la tarjeta.

```
import Connection from '../../../conections/Connection'
...
...
// Create card
var card =
{
  idCard: 0,
  mode:"CcTest",
  data: {
    questionText: question,
    optionsAnswer: arrOptionsAnswer,
  }
}
// Save Card
Connection.SaveCard(idCourse, idUnit, card).then()
...
...
```

Tabla 6-14: Fichero CCreateCardTest.vue

Si apreciamos el fragmento de código de la *Tabla 6-14* la tarjeta es almacenada mediante la función *SaveCard* del módulo previamente importado *Conection*. Esta función recibe el identificador del curso para el que se está creando la tarjeta, el identificador de la unidad del curso y la tarjeta.

Por último, hay que indicar que este archivo debería denotarse por “*CCreate*” + nombre del nuevo tipo de tarjeta y generarse dentro del directorio `src/components` para continuar con la convención de nombre y estructura de directorios descrita en *apartado 6.8*.

6.13.3.2 Archivo de tarjeta

El segundo archivo sería la tarjeta de aprendizaje en sí, es decir como vimos en la *Ilustración 6-20* para las tarjetas de tipo test e *Ilustración 6-22* para las tarjetas de tipo video con subtítulos, este contiene las acciones que el usuario realiza con el objetivo de aprender o repasar el contenido.

Este archivo deberá contener la lógica de la tarjeta. Es decir, deberá incluir si nuestra tarjeta es de tipo test el comportamiento cuando el usuario interactúa con una pregunta seleccionando la opción errónea o correcta.

Independientemente del tipo de tarjeta y su funcionalidad, todas ellas deberán implementar un método común *CardCompleted* y que podemos apreciar en la *Tabla 6-15*. Este es el encargado de informar mediante un bus de eventos al resto del sistema que este suscrito al evento que la tarjeta ha sido completada por el usuario.

```
import {eventBus} from '../../event-bus'
...
...
// Method CardCompleted
CardCompleted () {
  ...
  ...
  //Advance and calculate next repeat
  eventBus.$emit('addDataProgress', this.countError)
  ...
  ...
}
...
...
```

Tabla 6-15: Método Completar tarjetas

Este evento que notifica que una tarjeta de aprendizaje ha sido completada, desencadenará principalmente dos acciones; en primer lugar, se encargará en función de número de errores cometidos y que facilitamos como parámetro de realizar todo lo referente con el cálculo del momento óptimo para volver

a repasar el contenido aprendido y almacenar dicha programación de repasos. En segundo lugar, se encargará de actualizar todos los indicadores de progreso y avanzar a la siguiente tarjeta de aprendizaje.

Como podemos apreciar en el proceso descrito, todo lo referente con el algoritmo *SM-2* para el cálculo del momento óptimo se encuentra aislado de la lógica de cada tipo de tarjeta de aprendizaje lo que posibilita que sea sencillo el añadir nuevas modalidades de tarjetas a la aplicación.

Por último, hay que indicar que al igual que el archivo de creación, este debería denotarse por *“CC” + nombre del nuevo tipo de tarjeta* y generarse dentro del directorio `src/components` para continuar con la convención de nombre y estructura de directorios descrita en *apartado 6.8*.

6.14 Despliegue

Para completar con este Trabajo Fin de Grado se ha procedido al despliegue de la aplicación. En los siguientes apartados de esta sección hablaremos sobre el despliegue del *backend* y la aplicación en diferentes plataformas y dispositivos.

6.14.1 El backend

Como vimos en la *sección 4.6* el *backend* de nuestra aplicación es *Firebase un Backend-as-a-Service* y que se encuentra desplegado desde el inicio del proyecto. Este *backend* está disponible para todos los clientes de las diferentes plataformas y que utilizarán la misma instancia de la base de datos *Cloud firestore* de la que hablamos en la *sección 4.7.2*.

6.14.2 El frontend en diferentes plataformas y dispositivos

Como desde un inicio se pretendió, el objetivo es que la aplicación resultante estuviese disponible para las principales plataformas y dispositivos. En los siguientes puntos de este apartado, mostraremos una visualización de la aplicación *Lembra* en las diferentes plataformas y dispositivos.

6.14.2.1 Aplicación Windows

Mediante el *framework Electron* del que hablamos en la *sección 4.4.2* la aplicación es compilada para plataformas *Windows*. En la *Ilustración 6-27* podemos apreciar el comando que nos permite compilar la aplicación para esta plataforma en versión de 64 bits.

```
quasar build -c --mode electron --target win32 --arch x64 --theme mat
```

Ilustración 6-27: Compilación plataformas Windows

El resultado de la compilación es un directorio el cual contiene 117 archivos y 2 subdirectorios entre los que se encuentran las diferentes librerías y ficheros necesarios para el funcionamiento de la aplicación *Windows*. Entre estos archivos se encuentra el fichero *Lembra.exe*, archivo ejecutable y que da acceso a la aplicación. Podemos apreciar el resultado de la ejecución la aplicación en una maquina con el sistema operativo Windows 10 en la *Ilustración 6-28*.

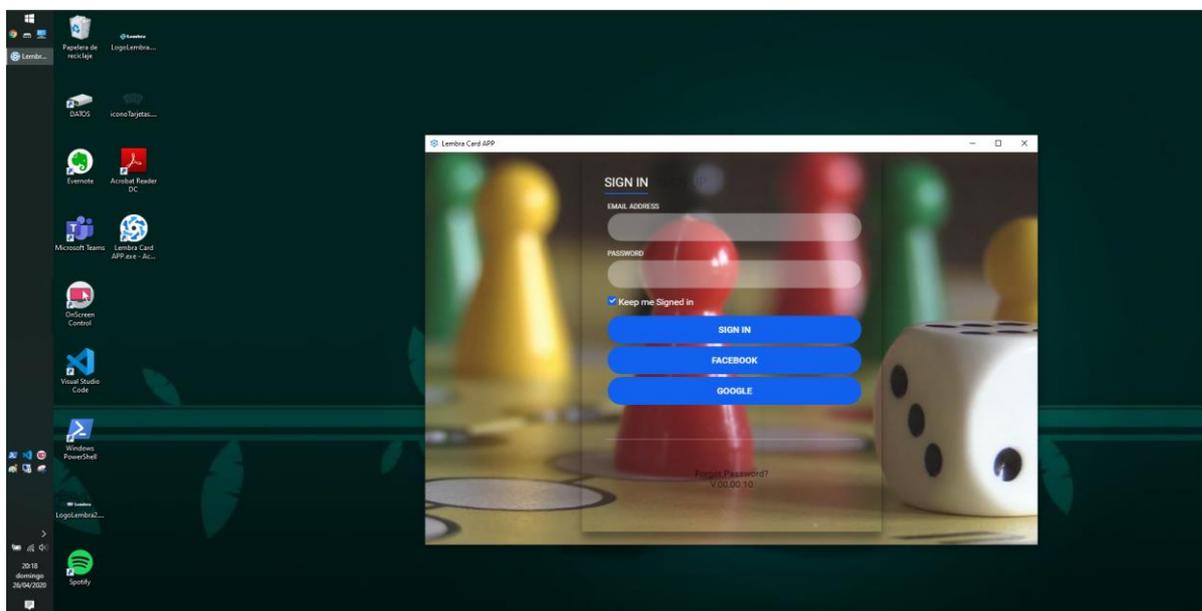


Ilustración 6-28: Lembra en Windows 10.

6.14.2.2 Aplicación Linux

Mediante el *framework Electron* del que hablamos en la *sección 4.4.2* la aplicación es compilada para plataformas *Linux*. En la *Ilustración 6-27* podemos apreciar el comando que nos permite compilar la aplicación para esta plataforma en versión de 64 bits.

```
quasar build -c --mode electron --target linux --arch x64 --theme mat
```

Ilustración 6-29: Compilación Linux

El resultado de la compilación es un directorio el cual contiene 71 archivos y 2 subdirectorios entre los que se encuentran las diferentes librerías y ficheros necesarios para el funcionamiento de la aplicación *en Linux*. Entre estos archivos se encuentra el fichero *Lembra* de tipo *application/x-executable*, archivo ejecutable y que da acceso a la aplicación. Podemos apreciar el resultado de la ejecución la aplicación en una maquina con el sistema operativo *Ubuntu* en la *Ilustración 6-30*.

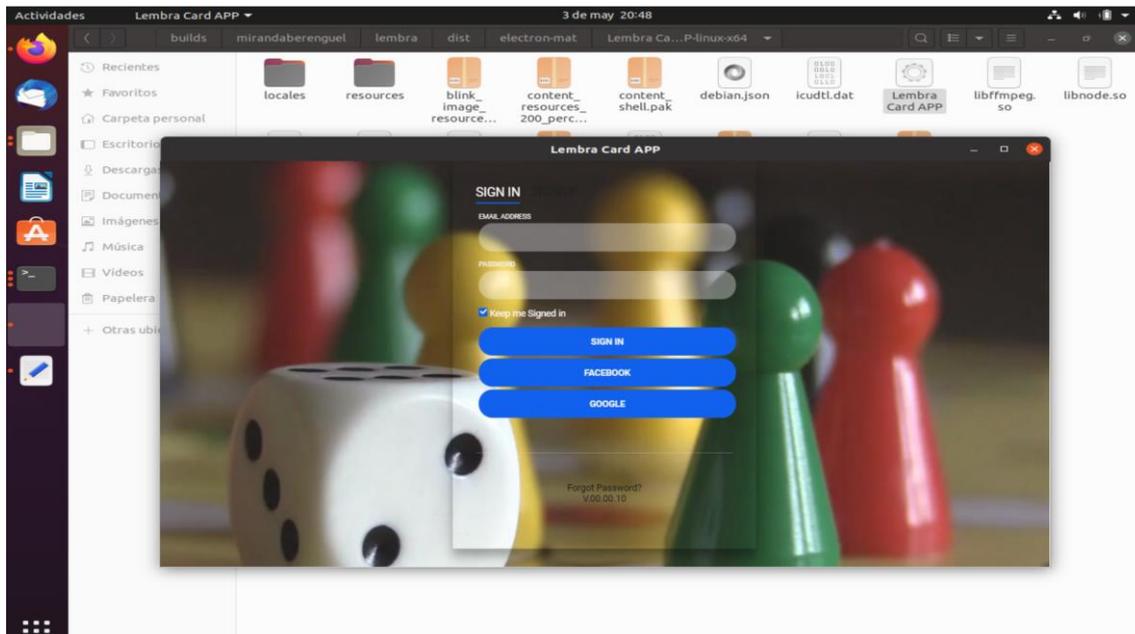


Ilustración 6-30: Lembra en Ubuntu

6.14.2.3 Aplicación Android

Mediante el *framework Apache Cordova* del que hablamos en la *sección 4.4.1* la aplicación es compilada para plataformas *Android*. En la *Ilustración 6-31* podemos apreciar el comando que nos permite compilar la aplicación para *Android*.

```
quasar build -m cordova -T android
```

Ilustración 6-31: Compilación Android

Este comando dará como resultado un archivo con extensión *.apk* (*Paquete de Aplicación Android*). Este formato es una variante del formato *JAR* de *Java* y se usa para distribuir e instalar componentes empaquetados para la plataforma *Android*. En la *Ilustración 6-32* podemos ver la aplicación *Lembra* funcionando en un emulador de un dispositivo móvil *Motorola Moto G4* con el sistema operativo *Android*.



Ilustración 6-32: Lembra en dispositivo Android

6.14.2.4 Aplicación IOs

Mediante el *framework* Apache Cordova del que hablamos en la *sección 4.4.1* la aplicación puede ser compilada para plataformas IOs. En la *Ilustración 6-33* podemos apreciar el comando que nos permite compilar la aplicación para IOs.

```
quasar build -m cordova -T ios
```

Ilustración 6-33: compilación IOs

La construcción de la aplicación debe realizarse obligatoriamente sobre un sistema operativo *macOS*⁷⁰ requiriéndose además las herramientas de *Apple* para desarrolladores, esto requiere disponer de una suscripción al *Apple Developer Program*⁷¹ es por esto para finalizar este trabajo se ha optado únicamente por mostrar la aplicación simulada en un dispositivo *iPhone 8* mediante las herramientas para desarrolladores de *Google Chrome*. Podemos ver esta simulación en la *Ilustración 6-34*.

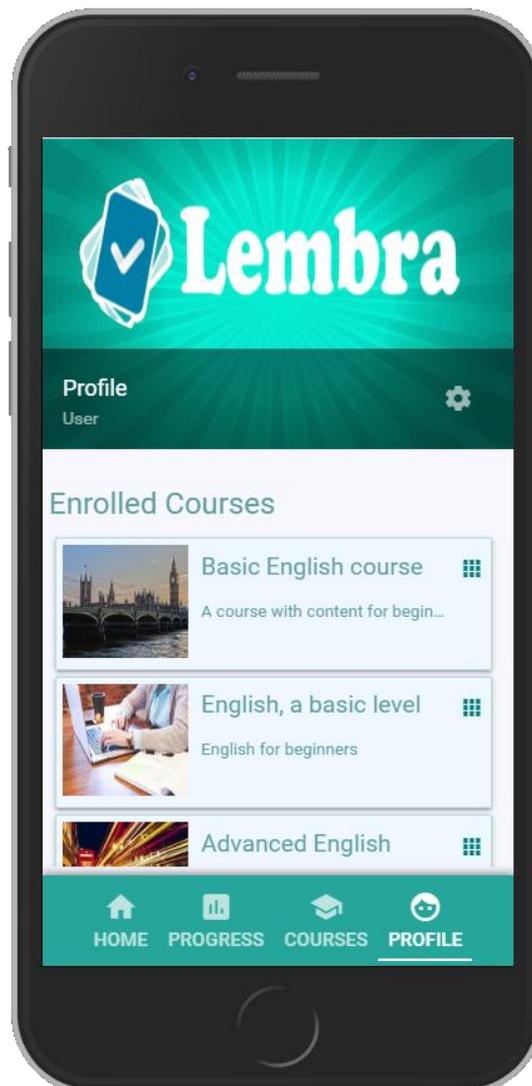


Ilustración 6-34: Lembra en iPhone 8

⁷⁰ **MacOS:** es el nombre del sistema operativo creado por Apple para su línea de computadoras Macintosh.

⁷¹ **Apple Developer Program:** permite firmar y distribuir las aplicaciones en la App Store. Esta suscripción tiene un coste anual.

6.14.2.5 Aplicación Web PWA

Para realizar el despliegue de la Aplicación Web Progresiva (PWA) directamente en hosting de *Firebase*, utilizamos su SDK⁷², el cual nos permite mediante un comando desplegar la aplicación. Podemos ver el comando utilizado para el despliegue en la siguiente ilustración.

```
firebase deploy --non-interactive --token ""
```

Ilustración 6-35: Despliegue en Firebase Hosting

La aplicación web progresiva (PWA), se encuentra accesible y disponible bajo la siguiente url: <https://lembra.mirandaberenguel.es>. Podemos apreciar su funcionamiento en la *Ilustración 6-36*

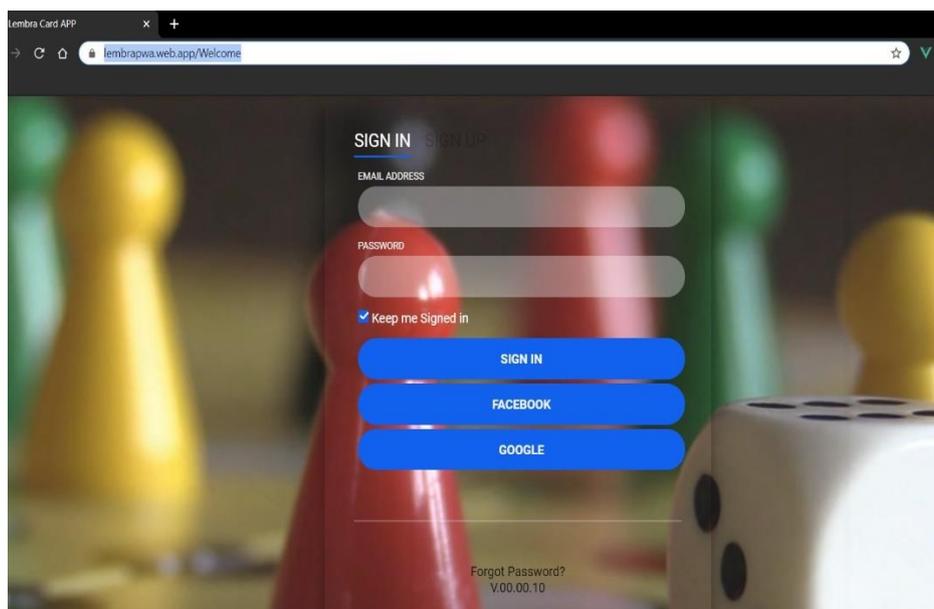


Ilustración 6-36: PWA Lembra alojada en Firebase hosting

Uno de los aspectos que nos hace identificar *Lembra* como una PWA frente a una SPA (aplicación de una sola página) u otro tipo de web es que, tras su acceso por primera vez, ya sea desde un dispositivo móvil o a través de un navegador web de escritorio, esta nos indicara que está disponible para su instalación. Podemos ver esto en la *Ilustración 6-37*.

⁷² **SDK**: es un conjunto de herramientas de desarrollo de software que permite a un desarrollador de software crear una aplicación informática para un sistema concreto.

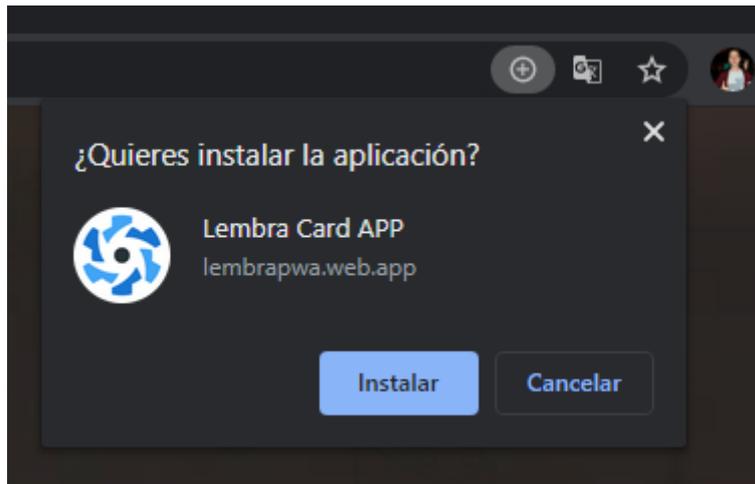


Ilustración 6-37: Lembra PWA.



Capítulo

7

PRUEBAS y CONTROL DE CALIDAD

7.1 Introducción

El control de calidad es algo que debemos tener en cuenta en cualquier desarrollo software y al que en muchas ocasiones no dedicamos suficientes recursos de los empleados para el desarrollo del proyecto. En ocasiones los procesos de prueba pueden resultar costosos y laboriosos, pero se hacen imprescindibles para garantizar que el software resultante cumple unos mínimos parámetros de calidad [42]. Entre las pruebas, no solo se debe comprobar el correcto funcionamiento del producto, sino que, también se hace necesario comprobar la facilidad de uso, la **accesibilidad**⁷³ del software y su **usabilidad**⁷⁴.

El objetivo de este capítulo es describir el proceso seguido para generar **casos de prueba** e informes de la aplicación resultante que nos han ofrecido datos sobre la **calidad del producto** y nos han permitido tomar decisiones que han mejorado la calidad y la corrección de errores de forma clara y efectiva.

⁷³ **Accesibilidad:** es el grado en el que las personas pueden utilizar o acceder a un servicio, independientemente de sus capacidades técnicas, cognitivas o físicas.

⁷⁴ **Usabilidad:** se refiere a la facilidad con que las personas pueden utilizar un software.

7.2 Plan de Pruebas

En este apartado se pretende definir el plan de pruebas funcionales que permitan validar los casos de uso definidos en el *capítulo 5*, verificándose que los requisitos han sido cumplidos. Para la documentación de los requisitos, se ha optado por utilizar una simplificación de la estructura de plantillas propuestas por el **Marco de desarrollo de la Junta de Andalucía** (MADEJA).

7.2.1 Definición de los casos de prueba

CP-001	<i>Registrarse</i>
Descripción: Realizar un registro de un nuevo usuario en la aplicación.	
Prerrequisitos: Ninguno.	
Resultado esperado: El sistema deberá comportarse, registrando y autenticando al nuevo usuario dándole acceso a la aplicación.	

CP-002	<i>Registrarse con un mail ya registrado en el sistema.</i>
Descripción: Intentar realizar el registro en la aplicación mediante un email ya registrado.	
Prerrequisitos: Ninguno.	
Resultado esperado: El sistema deberá comportarse informando al usuario que intenta registrarse que el mail indicado ya se encuentra registrado en la aplicación.	

CP-003	<i>Autenticarse en el sistema.</i>
Descripción: Realizar una autenticación en el sistema con el usuario test, mediante su email y contraseña.	
Prerrequisitos: Ninguno.	
Resultado esperado: El sistema deberá comportarse autenticando al usuario y dándole acceso a la aplicación.	

CP-004	<i>Visualizar la página de inicio (Home)</i>
Descripción: Visualizar la página Home del menú de navegación.	
Prerrequisitos: El usuario ha tenido que autenticarse.	
Resultado esperado: El sistema deberá comportarse mostrando al usuario la página Home con el resumen del curso actual matriculado.	

CP-005	<i>Visualizar la página de Estadísticas.</i>
---------------	-----------------------------------------------------

Descripción: Visualizar la página Estadísticas del menú de navegación.
Prerrequisitos: El usuario ha tenido que autenticarse.
Resultado esperado: El sistema deberá comportarse mostrando al usuario la página con las estadísticas diarias y semanales del curso matriculado.

CP-006	<i>Visualizar la página de Cursos.</i>
Descripción: Visualizar la página Cursos del menú de navegación.	
Prerrequisitos: El usuario ha tenido que autenticarse.	
Resultado esperado: El sistema deberá comportarse mostrando al usuario la página con un listado de los cursos disponibles del sistema, catalogados como públicos, mostrándose de los datos del curso, nombre, descripción, imagen principal, y si el usuario está matriculado en el curso.	

CP-007	<i>Visualizar la página de perfil y configuraciones.</i>
Descripción: Visualizar la página de perfil y configuraciones del menú de navegación.	
Prerrequisitos: El usuario ha tenido que autenticarse.	
Resultado esperado: El sistema deberá comportarse mostrando al usuario la página con los siguientes datos del usuario: email, tipo de registro. El sistema muestra un listado de los cursos matriculados y otro con los creados por el usuario, mostrando los siguientes datos: nombre del curso y descripción. El sistema muestra un listado de las opciones de configuración establecidas por el usuario permitiendo cambiarse.	

CP-008	<i>Mostrar el detalle de un curso.</i>
Descripción: Visualizar el detalle de un curso, datos del curso y unidades del curso.	
Prerrequisitos: El usuario ha tenido que autenticarse y seleccionar un curso.	
Resultado esperado: El sistema deberá comportarse mostrando al usuario una página con los siguientes datos del curso: nombre, descripción, tipo e imagen. El sistema muestra un listado de las unidades del curso ordenadas por el campo orden, mostrando los siguientes datos: nombre, descripción y número de tarjetas de aprendizaje.	

CP-009	<i>Buscar cursos.</i>
Descripción: Realizar una búsqueda de cursos en el sistema.	
Prerrequisitos: El usuario ha tenido que autenticarse.	
Resultado esperado:	

El sistema deberá comportarse mostrando al usuario un listado de los cursos que cumplen el criterio de búsqueda, catalogados como públicos, mostrándose de los datos del curso, nombre, descripción, imagen principal, y si el usuario está matriculado en el curso.

CP-010	<i>Crear un nuevo curso.</i>
Descripción: Realizar el alta de un nuevo curso.	
Prerrequisitos: El usuario ha tenido que autenticarse.	
Resultado esperado: El sistema deberá comportarse mostrando al usuario un formulario que permita introducir los siguientes datos del curso: nombre, descripción, imagen, tipo de curso. Si todos los datos son validados por el sistema, deberá mostrarse un mensaje al usuario informando que el curso fue creado correctamente. La aplicación tras crear un curso correctamente debe redirigir al detalle del curso.	

CP-011	<i>Matricularse en un curso.</i>
Descripción: Matricularse en un curso de la aplicación.	
Prerrequisitos: El usuario ha tenido que autenticarse.	
Resultado esperado: El sistema deberá comportarse informando al usuario que se encuentra matriculado y dar la posibilidad de establecer el curso como curso principal.	

CP-012	<i>Desmatricularse en un curso.</i>
Descripción: Desmatricularse en un curso de la aplicación en el que el usuario este matriculado.	
Prerrequisitos: El usuario ha tenido que autenticarse. El usuario debe estar matriculado en el curso. El curso del que desea desmatricularse no puede estar establecido como principal.	
Resultado esperado: El sistema deberá comportarse informando al usuario que se ha desmatriculado del curso y dar la posibilidad de volver a matricularse.	

CP-013	<i>Establecer curso como principal.</i>
Descripción: Establecer un curso de la aplicación como curso principal.	
Prerrequisitos: El usuario ha tenido que autenticarse. El usuario debe estar matriculado en el curso.	
Resultado esperado: El sistema deberá comportarse informando al usuario que se ha establecido el curso como principal.	

CP-014	<i>Modificar los datos de perfil y configuración</i>
---------------	-------------------------------------------------------------

Descripción: Modificar los datos como nombre de usuario, o email, contraseña y los datos de configuración de la aplicación.
Prerrequisitos: El usuario ha tenido que autenticarse.
Resultado esperado: El sistema deberá comportarse informando al usuario de los datos de su cuenta: nombre, email y contraseña y dando la posibilidad de modificarlos. También debe ser posible modificar la configuración de usuario.

CP-015	<i>Modificar un curso y sus unidades.</i>
Descripción: Modificar los datos de un curso, incluyéndose: nombre, descripción del curso y sus correspondientes unidades del curso.	
Prerrequisitos: El usuario ha tenido que autenticarse. El usuario debe ser el propietario del curso.	
Resultado esperado: El sistema deberá comportarse permitiendo al usuario modificar los datos del curso, incluyendo las unidades que contenga el curso. Si las modificaciones son correctas, se informará al usuario que se guardaron correctamente.	

CP-016	<i>Eliminar un curso y sus unidades.</i>
Descripción: Dar de alta un nuevo curso y una unidad como contenido del curso. Eliminar la unidad y posteriormente eliminar el curso.	
Prerrequisitos: El usuario ha tenido que autenticarse.	
Resultado esperado: El sistema deberá comportarse permitiendo al usuario eliminar la unidad y el curso previamente creados.	

CP-017	<i>Crear, listar, modificar y eliminar tarjetas de aprendizaje.</i>
Descripción: Crear una tarjeta de aprendizaje dentro de la unidad de un curso y posteriormente listar el número de tarjetas, editar los datos de la tarjeta creada. Para finalizar eliminar la tarjeta de aprendizaje creada.	
Prerrequisitos: El usuario ha tenido que autenticarse. El usuario tiene que ser el propietario del curso.	
Resultado esperado: El sistema deberá comportarse permitiendo crear una nueva tarjeta de aprendizaje, listar el número de tarjetas de aprendizaje creadas y permitir eliminarla mostrando una confirmación de la eliminación.	

CP-018	<i>Listar y editar todos los cursos del sistema.</i>
Descripción: Acceder a la aplicación con un usuario con perfil de SuperUsuario y listar todos los cursos del sistema e intentar editar un curso no creado por SuperUsuario. Desautenticarse con el perfil de SuperUsuario y autenticarse con un usuario demo, el sistema no debe mostrar los cursos	

del sistema para su edición a excepción de los cursos creados por el usuario demo.
Prerrequisitos: El usuario ha tenido que autenticarse.
Resultado esperado: Con el SuperUsuario el sistema deberá comportarse permitiendo listar todos los cursos del sistema y permitiendo su edición aun no siendo el usuario que acceder a la aplicación el propietario del curso. Con el usuario demo, el sistema no debe permitir la edición de los cursos no creados por este usuario.

CP-019	<i>Practicar tarjetas de aprendizaje.</i>
Descripción: Seleccionar una unidad de un curso que contenga tarjetas de aprendizaje para practicar y completar todas las tarjetas de aprendizaje de la unidad.	
Prerrequisitos: El usuario ha tenido que autenticarse.	
Resultado esperado: El sistema obtiene las tarjetas de aprendizaje de la unidad y las muestra al usuario individualmente y por orden. El usuario completará individualmente cada tarjeta de aprendizaje, hasta que el número de tarjetas de aprendizaje pendientes de completar sea cero. El sistema deberá guardar las informaciones sobre el progreso en cada tarjeta de aprendizaje y en función de los datos de errores y aciertos mostrar esta información en el apartado estadísticas de la aplicación.	

CP-020	<i>Repasar tarjetas de aprendizaje.</i>
Descripción: Seleccionar una unidad de un curso que contenga tarjetas de aprendizaje para practicar y completar todas las tarjetas de aprendizaje de la unidad cometiendo al menos un fallo en cada tarjeta de aprendizaje. Tras finalizar todas las tarjetas de la unidad, en el menú acceder a repasar.	
Prerrequisitos: El usuario ha tenido que autenticarse.	
Resultado esperado: El sistema deberá mostrar todas las tarjetas de aprendizaje de la unidad como tarjetas pendientes de repasar.	

7.2.2 Trazabilidad de los casos de prueba

En este apartado se pretende aportar una matriz, en la cual se indicará la correspondencia entre los casos de pruebas definidos, y los casos de uso de la especificación de requisitos descritos en el apartado 5.4.2.3. En la matriz, las columnas representan los casos de uso y las filas cada uno de los casos de prueba. Si un caso de prueba se encarga de verificar un requisito, la casilla correspondiente estará marcada por un punto. Podemos ver la matriz en la *Tabla 7-1* la cual se encuentra girada horizontalmente para su correcta visualización de forma completa en una sola página.

7.3.2 Métricas de Accesibilidad

Al hablar de **accesibilidad**, se está haciendo referencia a un diseño que va a permitir que personas con algún tipo de discapacidad puedan percibir, entender, navegar e interactuar con la aplicación, aportando a su vez contenidos. La accesibilidad Web también beneficia a otras personas, incluyendo personas de edad avanzada que han visto mermadas sus habilidades a consecuencia de la edad. En este análisis se realiza una medición del grado de accesibilidad de la aplicación resultante, obteniendo una puntuación total de 81% como podemos ver en la *Ilustración 7-1*.

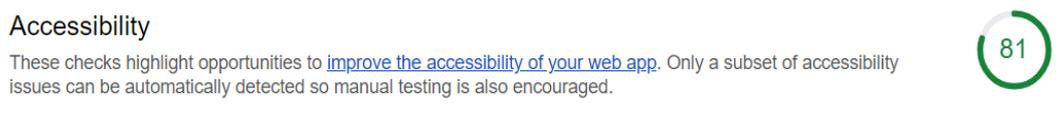


Ilustración 7-1: Resultados Accesibilidad Aplicación

Debemos tener en cuenta, como en el propio informe se indica, que hay muchos aspectos que no se pueden medir por parte de Google *Lighthouse* y que debemos revisar manualmente. Dentro de los criterios que sí se han medido, existen algunos que fueron superados y otros aún están pendientes de realizar cambios para alcanzar un mayor nivel de accesibilidad. Detallamos los que consideramos de mayor relevancia en la *Tabla 7-2*.

Datos de Auditoria	Estado
<p>Botones tienen un nombre accesible</p> <p>Cuando un botón no tiene un nombre accesible, los lectores de pantalla lo anuncian como "botón", lo que lo hace inutilizable para los usuarios que confían en los lectores de pantalla.</p>	✓
<p>Los elementos describen bien los contenidos</p> <p>Los contenidos serán más fáciles de entender para un usuario de tecnología asistencial, como un lector de pantalla</p>	⚠
<p>Los elementos del formulario no tienen etiquetas asociadas</p> <p>Las etiquetas aseguran que los controles de formulario sean anunciados correctamente por las tecnologías de asistencia, como los lectores de pantalla</p>	⚠
<p>Enlaces tienen un nombre discernible</p> <p>El texto del enlace (y el texto alternativo para las imágenes, cuando se utiliza como enlaces) que es discernible, único y enfocable mejora la experiencia de navegación para los usuarios de lectores de pantalla</p>	✓
<p>El contraste de color es satisfactorio</p> <p>Los colores de fondo y primer plano tienen una relación de contraste suficiente</p>	✓
<p>La página especifica un idioma.</p> <p>Si una página no especifica un atributo <i>lang</i>, un lector de pantalla asume que la página está en el idioma predeterminado que el usuario eligió al configurar el lector de pantalla. Si la página no</p>	✓

está en el idioma predeterminado, es posible que el lector de pantalla no anuncie el texto de la página correctamente	
-----------------------------------------------------------------------------------------------------------------------	--

Tabla 7-2: Datos auditoria accesibilidad

7.3.3 Métricas de tiempo de actuación

Estas métricas realizan un análisis de la aplicación principalmente en su carga inicial, detectando cargas lentas por archivos o imagen demasiado pesados y que nos han permitido identificar varios problemas de carga lenta que han sido solventados, obteniendo en las etapas finales del proyecto un porcentaje de 81%.

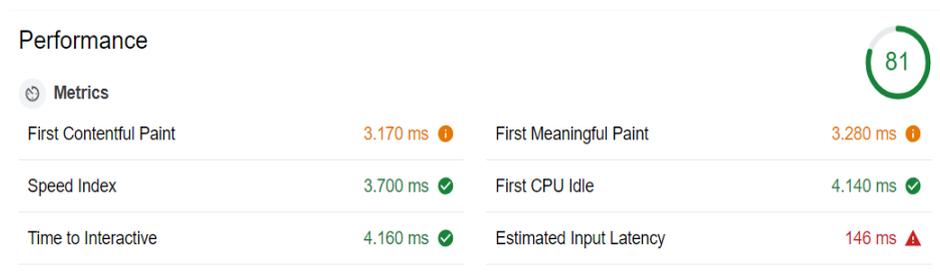


Ilustración 7-2: Métricas tiempos de actuación

Aun siendo un valor alto y que se encuentra catalogado como bueno dentro de la propia escala de *Lighthouse*, aún existen problemas por solventar y otros ya resueltos como los que vemos en la *Tabla 7-3*.

Datos de Auditoria	Ahorro estimado
Sirve imágenes en formatos de próxima generación Los formatos de imagen como JPEG 2000, JPEG XR y WebP a menudo proporcionan una mejor compresión que PNG o JPEG, lo que significa descargas más rápidas y menos consumo de datos.	0,75 s
Deferir CSS sin usar Elimine las reglas no utilizadas de las hojas de estilo para reducir los bytes innecesarios consumidos por la actividad de la red.	0,3 s
Utiliza una política de caché ineficiente en activos estáticos Una larga vida útil de caché puede acelerar las visitas repetidas a su página	
JavaScript tiempo de arranque Considere reducir el tiempo dedicado a analizar, compilar y ejecutar JS.	1.980 ms
Imágenes de tamaño adecuado.	

Sirve imágenes de tamaño adecuado para guardar datos móviles y mejorar el tiempo de carga	
Minificar CSS y JavaScript La reducción de los archivos de JavaScript y CSS puede reducir el tamaño de la carga útil y el tiempo de análisis de secuencia de comandos.	✓

Tabla 7-3: Datos auditoria tiempos de actuación

7.3.4 Métricas sobre el uso de buenas prácticas

Existen una serie de reglas en el desarrollo web que ayudan a mejorar la calidad de las aplicaciones resultantes, el análisis realizado por Google *Lighthouse* nos da una visión sobre el uso de alguna de las mejores practicas empleadas y sobre el que obtenemos una puntuación de 93%.

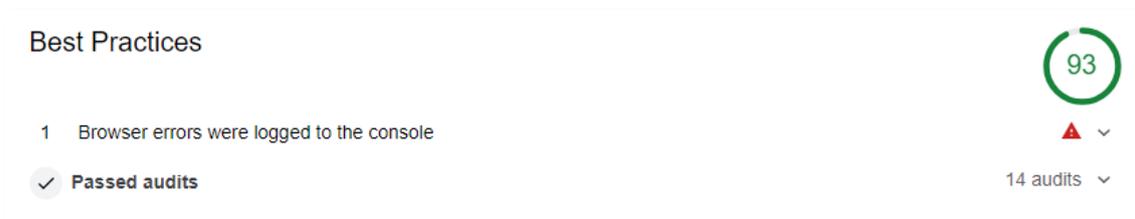


Ilustración 7-3 Métricas sobre el uso de buenas prácticas

En la *Tabla 7-4* detallamos algunas de las auditorías realizadas y que han dado lugar a la puntuación otorgada en el análisis como resultado final.

Datos de Auditoria	Estado
Evita la caché de la aplicación Application Cache está en desuso	✓
Evita la base de datos WebSQL Web SQL está en desuso. Considere usar IndexedDB en su lugar.	✓
Evita solicitar el permiso de geolocalización en la carga de la página Los usuarios desconfían o confunden a los sitios que solicitan su ubicación sin contexto	✓
Evita las bibliotecas de JavaScript front-end con vulnerabilidades de seguridad conocidas. Algunos scripts de terceros pueden contener vulnerabilidades de seguridad conocidas que son fácilmente identificadas y explotadas por los atacantes	✓
La página tiene el tipo de documento HTML La especificación impide que el navegador cambie a modo peculiar.	✓
Evita las API obsoletas Las API obsoletas eventualmente se eliminarán del navegador.	✓

<p>Permite a los usuarios pegar en campos de contraseña La prevención del pegado de contraseñas socava la buena política de seguridad</p>	
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------

Tabla 7-4: Datos auditoria buenas prácticas

7.3.5 Métricas aplicación Web Progresiva

Como vimos en la *sección 4.2* de este trabajo las PWA son aplicaciones web que pueden parecer al usuario una aplicación móvil nativa. Pero además de esto, se deben cumplir varias características, que fueron detalladas con anterioridad y que han sido verificadas en el informe que Google *Lighthouse* a realizado y en el que obtenemos una puntuación final de 100% como vemos en la siguiente ilustración:

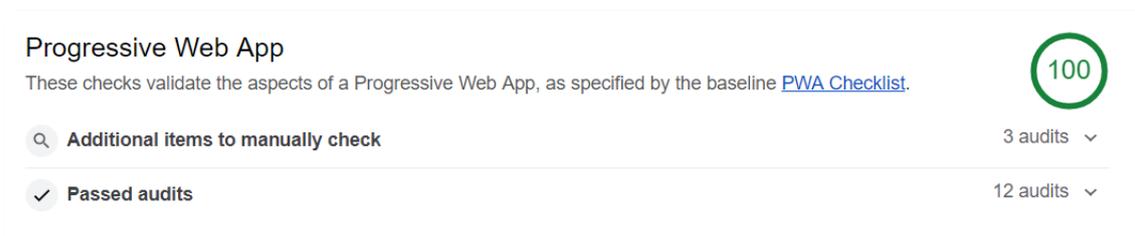


Ilustración 7-4: Métricas aplicación web progresiva

Para detallar, algunas de las auditorías realizadas por *Google Lighthouse*, incluimos la siguiente tabla donde son detalladas las que consideramos más importantes.

Datos de Auditoria	Estado
<p>La carga de página es lo suficientemente rápido en 3G Una carga de página rápida a través de una red 3G garantiza una buena experiencia de usuario móvil.</p>	
<p>Responde con un código 200 cuando está fuera de línea Utilizar un trabajador de servicios para que su aplicación pueda funcionar sin conexión.</p>	
<p>Se puede solicitar al usuario que instale la aplicación web Los navegadores pueden solicitar de manera proactiva a los usuarios que agreguen su aplicación a su pantalla de inicio, lo que puede generar un mayor compromiso.</p>	
<p>La barra de direcciones coincide con los colores de la marca La barra de direcciones del navegador puede ser temática para que coincida con su sitio.</p>	
<p>Registra a un trabajador de servicio El trabajador del servicio es la tecnología que permite que su aplicación use muchas funciones de la aplicación <i>Progressive Web App</i>,</p>	

como sin conexión, agregar a la pantalla de inicio y notificaciones push	
Contiene algo de contenido cuando JavaScript no está disponible Su aplicación debería mostrar algo de contenido cuando JavaScript está desactivado, incluso si solo es una advertencia para el usuario de que JavaScript es necesario para usar la aplicación	✓
El contenido tiene el tamaño correcto para la ventana gráfica Si el ancho del contenido de su aplicación no coincide con el ancho de la ventana gráfica, es posible que su aplicación no esté optimizada para pantallas móviles	✓

Tabla 7-5: Datos auditoria PWA

7.3.6 Métricas de posicionamiento SEO

Por el momento la aplicación no está disponible para un uso público, pero es importante que si, en algún momento se opta por su publicación se tengan en cuenta criterios de construcción que permitan el posicionamiento en buscadores y motores de búsqueda. Es por esto por lo que *Google Lighthouse* aporta un informe sobre algunas métricas obtenidas de la aplicación y en el que obtenemos de 90% como mostrados en la *Ilustración 7-5*.

SEO

These checks ensure that your page is optimized for search engine results ranking. There are additional factors Lighthouse does not check that may affect your search ranking. [Learn more](#).



Ilustración 7-5: Métricas parámetros para el posicionamiento SEO



Capítulo

8

CONCLUSIONES Y TRABAJO FUTURO

8.1 Introducción

En cualquier proyecto de cualquier disciplina, sacar conclusiones es la última parte, pero no la menos importante del proceso ya que la conclusión es fundamental para determinar el éxito o el fracaso del proyecto. De todos modos, sea cual sean los resultados, considero que todo el conocimiento adquirido en la realización de este trabajo fin de grado nunca habrá sido tiempo perdido, sino una gran inversión para mi futuro personal y profesional. ¡Sigue aprendiendo y explorando!

En este capítulo se comentarán las conclusiones del autor y las posibles líneas de mejora y sobre las que considero son interesantes continuar un trabajo futuro dentro de la misma línea del proyecto.

8.2 Conclusiones

Como vimos en el *capítulo 1* de este trabajo fin de grado, al comienzo de este se estableció como objetivo principal la construcción de una aplicación. Para cometer con este objetivo principal se comenzó por analizar las preferencias de los usuarios, este análisis se realizó en el *capítulo 3* y nos permitió decidirnos por la construcción de una aplicación multiplataforma y multidispositivo que permitiese cubrir el mayor número de usuarios posibles. Además, previo a la construcción fueron analizadas diferentes tecnologías y herramientas siendo justificada su elección frente a otras tal y como fue expuesto en el *capítulo 4*, todo esto junto con la aplicación de la metodología de trabajo defendida en el *capítulo 2* y el plan de pruebas y control de calidad expuesto en el *capítulo 7* han dado como resultado una aplicación funcional.

Es por lo expuesto con anterioridad y como autor del proyecto considero que los objetivos principales han sido cumplidos, se ha trabajado con una idea y se ha llevado a cabo. Podemos afirmar que la aplicación resultante es usable aun siendo conscientes que muchas de las funcionalidades no están finalizadas debido a las limitaciones que supone realizar, documentar y defender en tiempo y forma este trabajo.

Como objetivos secundarios y no menos importantes se marcó la realización de este trabajo en un tiempo finito y definido antes del comienzo de este. Quizás esta es la parte en la que no se han cumplido en su totalidad los objetivos propuestos y planificados en un inicio ya que por mi actividad profesional la finalización de este trabajo fin de grado se ha visto retrasada en el tiempo.

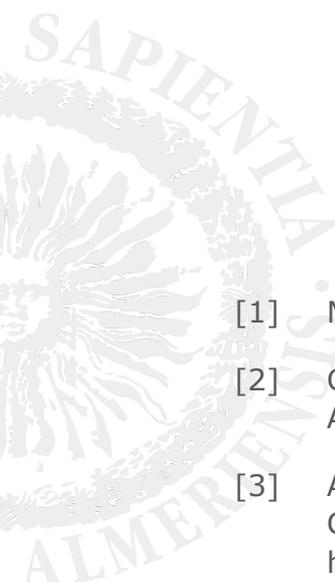
Por último, y no menos importante he de indicar que este trabajo fin de grado me ha permitido reflejar las experiencias y conocimientos adquiridos durante los más de diez años que han transcurrido desde que un día decidí comenzar a estudiar informática. Durante todo el proyecto se ha intentado defender y justificar la elección de las metodologías y herramientas utilizadas, contrastando la información mediante artículos, libros y otras fuentes de información que hoy en día es difícil no utilizar como blogs, web o video tutoriales entre otras. Este contraste de información hace que este documento no solo sea un trabajo fin de grado que defiende los criterios del autor, sino que puede ser un apoyo futuro para nuevos proyectos.

8.3 Líneas de Mejora

Antes de comenzar el proyecto fin de grado fue necesario definir su alcance y limitaciones, en esta etapa quedaron diversas funcionalidades y aspectos fuera del alcance y que pueden ser consideradas como trabajo futuro dentro de la continuidad de este trabajo. Es por este motivo las líneas de trabajo futuro están orientadas en los siguientes aspectos:

- Uno de los objetivos cubiertos del proyecto ha sido el de generar una base sólida, desde la que poder obtener **gran cantidad de datos** sobre los hábitos de estudio y resultados de los usuarios. Como trabajo futuro se pretende realizar un análisis de los datos obtenidos a gran escala y que permitan generar **modelos de inteligencia artificial** que posibiliten para cada usuario generar un modelo personalizado que permita tomar la decisión sobre el momento óptimo para volver a repasar un contenido ya estudiado para el usuario. Esto permitiría reemplazar al modelo SM-2 empleado y común para todos los usuarios de la aplicación por modelos específicos y entrenados por y para cada usuario.
- Uno de los objetivos iniciales del proyecto y que fueron descartados por falta de tiempo es la inclusión de **servicios cognitivos digitales** que incorporen la capacidad de hablar, escuchar y comprender el texto escrito por los usuarios de la aplicación. Un ejemplo de esto puede ser la interpretación de la voz de los usuarios para convertirlo en texto y comprobar su pronunciación. Existen diversas *API's* de *Microsoft*, *IBM* o *Google* que posibilitarían realizar esta integración de una forma sencilla sin la necesidad de realizar desarrollos específicos en esta área de la inteligencia artificial.

Bibliografía

- 
- [1] M. Prensky, Enseñar a nativos digitales, Madrid: Ediciones SM, 2011.
- [2] C. PWA, «Comunidad Progressive Web Apps» 2019. [En línea]. Available: pwaexperts.io. [Último acceso: 15 08 2019].
- [3] Apache Software Foundation, «Official documentation Apache Cordova» [En línea]. Available: <https://cordova.apache.org/docs/en/latest/>. [Último acceso: 16 06 2018].
- [4] GitHub Inc., «Official documentation Electron Framework» [En línea]. Available: <https://electronjs.org/>. [Último acceso: 16 06 2018].
- [5] Junta de Andalucía, «Marco de Desarrollo de la Junta de Andalucía (MADEJA)» 01 03 2013. [En línea]. Available: <http://www.juntadeandalucia.es/servicios/madeja/contenido/libro-pautas/148>. [Último acceso: 03 04 2018].
- [6] P. K. Per Kroll, The rational unified process made easy a practitioner's guide to the RUP., Boston, MA: Addison-Wesley, 2003.
- [7] G. Pollice, L. Augustine, C. Lowe y J. Madhur, Software Development for Small Teams: A RUP-Centric Approach, Boston, Massachusetts.: Addison-Wesley Professional, 2003.
- [8] M. Wesch, «Una visión de los estudiantes de hoy» 2007. [En línea]. Available: <https://www.youtube.com/watch?v=dGCJ46vyR9o>. [Último acceso: 01 06 2018].
- [9] L. Tíscar, «Aprender desde los Márgenes, Mobile Learning para una Sociedad Red,» de *TEDx Talks*, Madrid, 2011.
- [10] T. Connolly, M. Stansfield y L. Boyle, Advancements for Multi-Sensory Human Computer Interfaces: Techniques and Effective Practices, Hershey, New York: Information science reference, 2009.
- [11] H. Ebbinghaus, Memory: A contribution to experimental psychology, Berlín, 1913.

- [12] D. P. Wozniak, «Supermemo» 01 05 1998. [En línea]. Available: <https://www.supermemo.com/english/ol/sm2.htm>. [Último acceso: 15 06 2018].
- [13] P. C. Brown, H. L. Roediger y M. A. McDaniel, *Make it Stick: The Science of Successful Learning*, Cambridge, Massachusetts, Estados Unidos.: Harvard University Press, 2014.
- [14] S. H. K. Kang, «Spaced Repetition Promotes Efficient and Effective Learning: Policy Implications for Instruction» *Policy Insights from the Behavioral and Brain Sciences*, vol. Vol. 3, nº 1, p. 12–19, 2016.
- [15] P. Woźniak, «Theoretical aspects of learning» [En línea]. Available: <https://www.supermemo.com/articles/theory.htm>. [Último acceso: 21 05 2018].
- [16] N. J. Cepeda, N. Coburn, D. Rohrer, J. T. Wixted, M. C. Mozer y H. Pashler, «Optimizing distributed practice theoretical analysis and practical implications,» *Psicología Experimental*, vol. 56, nº 4, pp. 236–246, 2009.
- [17] S. Leitner, *So lernt man lernen: Der Weg zum Erfolg*, Herder Verlag GmbH, 1972.
- [18] X.-L. Pham, G.-D. Chen, Thi-Huyen y N. Wu, «Card-based design combined with spaced repetition: A new interface for displaying learning elements and improving active recall,» *Computers & Education*, vol. 98, nº 1, pp. 142–156, 2016.
- [19] «SuperMemopedia» [En línea]. Available: <http://www.supermemopedia.com>. [Último acceso: 05 06 2018].
- [20] Supermemo, «Supermemo» 05 01 2005. [En línea]. Available: <https://www.supermemo.com/english/algsm11.htm>. [Último acceso: 15 06 2018].
- [21] Anki, «apps.ankiweb,» [En línea]. Available: <https://apps.ankiweb.net/docs/manual.html>. [Último acceso: 03 06 2018].
- [22] J. Garcia, «Educacion y nativos digitales» de *TEDx Talks*, Valencia, 2013.
- [23] R. Barrera, «Como acabar con una educación aburrida» de *TEDxSevilla*, Sevilla, 2016.
- [24] F. Pedró, *Por favor enciendan sus celulares, la clase empieza.*, TEDxTalks, Ed., 2017.

- [25] H. Research, *Digital Study Trends Survey*, McGraw-Hill Education, 2016.
- [26] D. Hume, *Progressive Web Apps*, Shelter Island, Nueva York: Manning Publications, 2017.
- [27] T. Ater, *Building Progressive Web Apps: Bringing the power of native to the browser*, Farnham, Reino Unido: O'Reilly UK Ltd., 2017.
- [28] E. You, «Blog Evan You» 20 10 2015. [En línea]. Available: <http://blog.evanyou.me>. [Último acceso: 10 07 2018].
- [29] Vue.org, «Comparison with Other Frameworks» [En línea]. Available: <https://vuejs.org/v2/guide/comparison.html>. [Último acceso: 10 07 2018].
- [30] S. Rause, «Results for js web frameworks benchmark,» 20 11 2017. [En línea]. Available: <https://www.stefankrause.net/js-frameworks-benchmark7/table.html>. [Último acceso: 10 06 2018].
- [31] D. Flanagan, *JavaScript the definitive guide.*, Sebastopol: O'Reilly Media, 2010.
- [32] D. Crockford, «Blog Crockford» 2001. [En línea]. Available: <http://www.crockford.com/javascript/javascript.html>. [Último acceso: 15 07 2018].
- [33] D. Cederholm y E. Marcotte, *Handcrafted CSS more bulletproof web design*, Berkeley: New Riders, 2010.
- [34] A. Evangelist, «apievangelist.com» 1 2 2013. [En línea]. Available: <http://apievangelist.com/2012/06/03/rise-of-mobile-backend-as-a-service-mbaas-api-stacks/>. [Último acceso: 1 5 2019].
- [35] Google., «Firebase,» Firebase, 2012. [En línea]. Available: www.firebase.google.com. [Último acceso: 1 10 2018].
- [36] Firebase, «Documentación Cloud Firestore» [En línea]. Available: <https://firebase.google.com/docs/firestore/?hl=es-419>. [Último acceso: 15 06 2018].
- [37] Firebase, «Cloud Firestore» Firebase, [En línea]. Available: <https://firebase.google.com/docs/firestore>. [Último acceso: 15 06 2019].
- [38] B. S. Scott Chacon, *Pro Git*, Nueva York: Apress, 2014.
- [39] V. Driessen, «nvie.com» 05 2010. [En línea]. Available: <https://nvie.com/posts/a-successful-git-branching-model/>. [Último acceso: 2018 07 04].

- [40] GitLab, «GitLab Docs,» GitLab, [En línea]. Available: https://docs.gitlab.com/ee/workflow/gitlab_flow.html. [Último acceso: 04 07 2018].
- [41] E. Pidoux, Git best practices guide [Recurso electrónico] : master the best practices of Git with the help of real-time scenarios to maximize team efficiency and workflow, Birmingham, UK: Packt Publishing, 2014.
- [42] S. A. Sanchez, M. A. U. Sicilia y D. G. Rodriguez, Ingeniería del software : un enfoque desde la guía SWEBOK, Madrid: GARCETA GRUPO EDITORIAL, 2011.
- [43] J. d. Andalucía, «MADEJA,» [En línea]. Available: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/361>. [Último acceso: 2019 05 08].
- [44] VueJS, «Vuejs.org,» [En línea]. Available: <https://vuejs.org/v2/guide/>. [Último acceso: 07 22 2018].
- [45] Mozilla Firefox, «MDN Web Docs Mozilla,» [En línea]. Available: https://developer.mozilla.org/es/docs/Web/Web_Components. [Último acceso: 05 06 2018].
- [46] P. Halliday, Vue.js 2 design patterns and best practices [Recurso electrónico] : build enterprise-ready, modular Vue.js applications with Vuex and Nuxt., Birmingham, UK : Packt Publishing, 2018.
- [47] A. Hernández Chillón, S. Feliciano Morales, J. García Molina y D. Sevilla Ruiz, «Visualización de Esquemas en Bases de Datos NoSQL basadas en documentos,» de *JORNADAS DE INGENIERÍA DEL SOFTWARE Y BASES DE DATOS (JISBD 2017)*, Sevilla, 2017.
- [48] ITNext, «How to Structure a Vue.js Project» [En línea]. Available: <https://itnext.io/how-to-structure-a-vue-js-project-29e4ddc1aeeb>. [Último acceso: 07 20 2018].
- [49] A. Kyriakidis, The majesty of Vue.js, Birmingham, UK : Packt Publishing, 2016.
- [50] S. Florian, M. Robert y V. Oliver, «SPACED REPETITION LEARNING GAMES ON MOBILE DEVICES FOUNDATIONS AND PERSPECTIVES» *Interactive Technology and Smart Education*, vol. 11, nº 3, pp. 201-222, 2014.
- [51] Google, «Lighthouse Google» Google, 11 11 2019. [En línea]. Available: <https://developers.google.com/web/tools/lighthouse>. [Último acceso: 2019 07 1].
- [52] K. Philippe, «ResearchGate» [En línea]. Available: https://www.researchgate.net/publication/31736942_The_Rational_

Unified_Process_An_Introduction_P_Kruchten. [Último acceso: 10 08 2018].



Apéndice



MANUAL DE USUARIO

A.1 Introducción

Lembra se ha diseñado con el objetivo de proporcionar a los usuarios la posibilidad de aprender diverso contenido de una forma fácil e intuitiva, porque la mejor forma de aprender es pasárselo bien mientras se aprende. Es por esto, Lembra fomenta el uso de contenido multimedia y la interacción con el usuario frente a la lectura de largos textos.

Lembra está pensada para estimular la mente, a diferencia de los libros de texto los cuales carecen de interacción, en Lembra es posible aprender con ejercicios variados en el momento adecuado e incluso crear nuestro propio contenido que podremos o no compartir con el resto de los usuarios de la comunidad Lembra.

En el siguiente capítulo se detalla un manual de usuario básico para comenzar a aprender mediante la aplicación o generar nuevo contenido.

A.2 Autenticación y registro en el sistema

Para acceder a la aplicación, el usuario debe de hacer uso de sus credenciales de acceso (email y contraseña). Por defecto al acceder a la aplicación aparecerá el formulario de autenticación <1> representado en la *Ilustración 1-1*. El usuario deberá introducir su email en el campo *email address* <3> y su contraseña en el campo *password* <4> por último deberá pulsar sobre el botón *sign in* <6>. En caso de que el usuario desee que los datos de acceso queden almacenados en el dispositivo podrá hacer uso el check *Keep Signed* <5>. Podemos apreciar lo anteriormente descrito en la *Ilustración 1-1*.

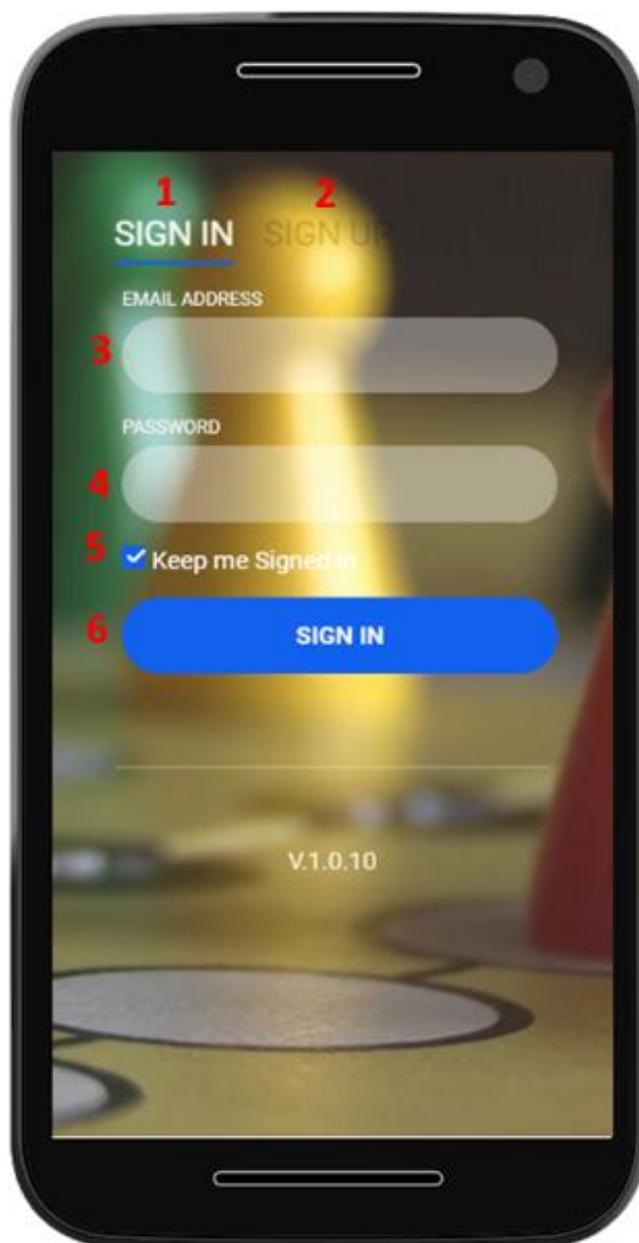


Ilustración 1-1: Pantalla de autenticación

En caso de que el usuario no disponga de credenciales de acceso deberá generar un nuevo usuario mediante el formulario de registro disponible y al que accederemos mediante el botón <1>, representado en la *Ilustración 1-2*. En este formulario se requerirá introducir un email valido en el campo *email address* <2> y una contraseña en el campo *password* <3> que por seguridad y para evitar posibles errores en la digitación se requerirá sea introducida por una segunda vez en el campo *repeat password* <4>. Tras esto se habilitará el botón <5> que podemos apreciar en la ilustración *Ilustración 1-2* y que nos dará acceso a la aplicación.

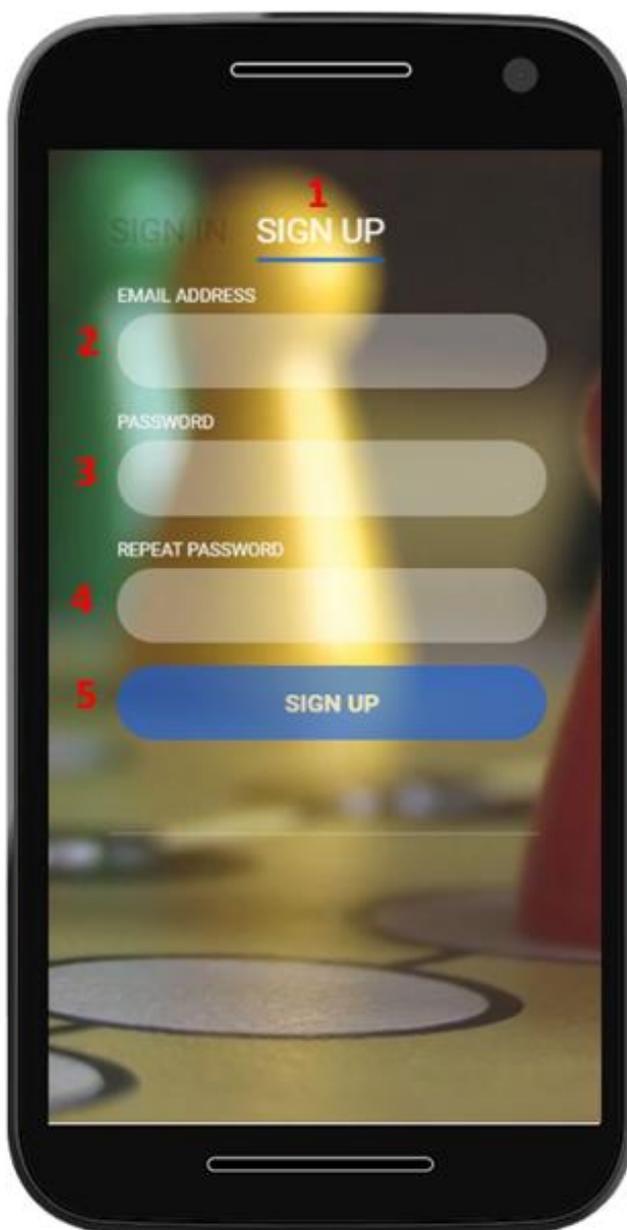


Ilustración 1-2: Pantalla de registro

A.3 Pantalla principal

Una vez autenticados en la aplicación nos aparecerá la pantalla principal. En ella se visualizarán los datos referentes al curso establecido como curso actual. Desde esta pantalla podemos continuar el aprendizaje mediante el botón *Continue Learning* <3> representado en la *Ilustración 1-3*. Además, en esta pantalla se visualizará si disponemos de tarjetas de aprendizaje pendientes de repasar mediante el icono <1> representado en la ilustración. Pulsado sobre este botón accederemos a las tarjetas pendientes de repaso, describiremos esto en la *sección A.3.1*.

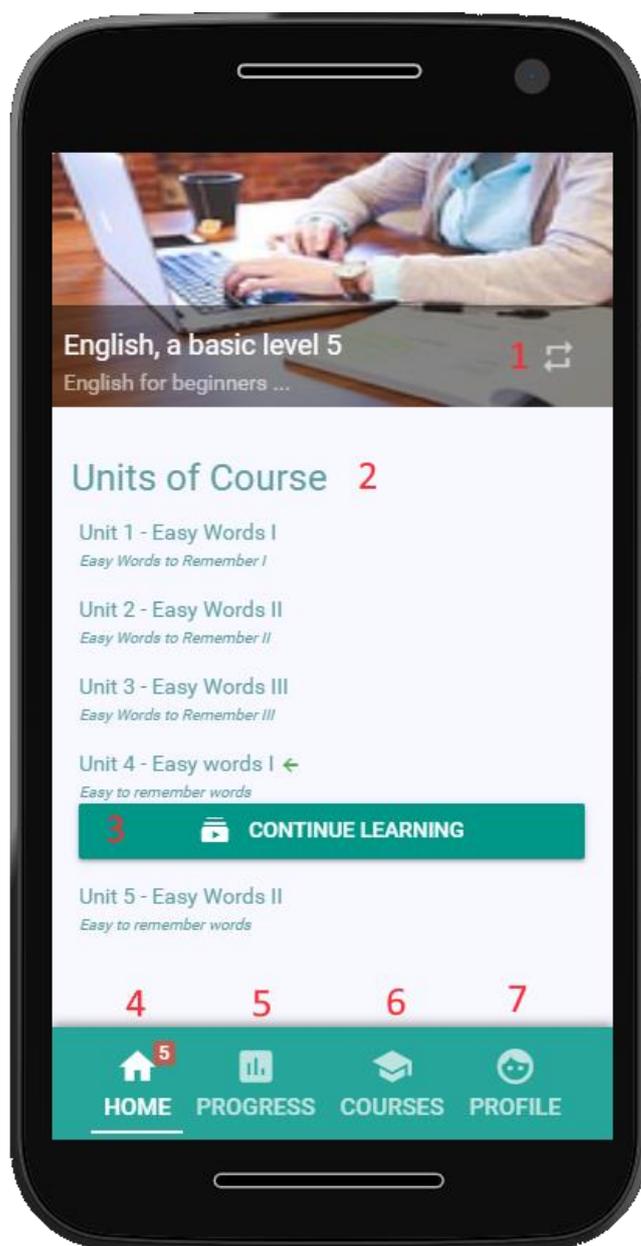


Ilustración 1-3: Pantalla principal

Además, desde esta pantalla tendremos acceso a la barra de navegación con las opciones *Home* <4>, *Progress* <5>, *Courses* <6>, *Profile* <7> siendo estas secciones de la aplicación que detallaremos en los siguientes apartados.

En la *Ilustración 1-4* podemos apreciar la representación de la pantalla principal en un dispositivo Tablet, PC o Web con una resolución de pantalla igual o superior a 1024 x 768 píxeles. Las opciones descritas con anterioridad se encuentran marcadas con la misma numeración, aunque como podemos apreciar debido a la adaptación en la representación del contenido estas pueden encontrarse ubicadas de forma diferente con el objetivo de mejorar su accesibilidad en este tipo de dispositivos.



Ilustración 1-4: Pantalla Home - PC y tablet

A.3.1 Modo aprendizaje y el modo repaso

En Lembra, los cursos están compuestos por diferentes unidades de aprendizaje. El progreso por las unidades de cada curso será almacenado de forma que podamos avanzar en el aprendizaje por las diferentes unidades de un curso. La unidad actual de aprendizaje se encontrará identificada por el botón <3> que da acceso a su contenido como podemos ver en la *Ilustración 1-5*.

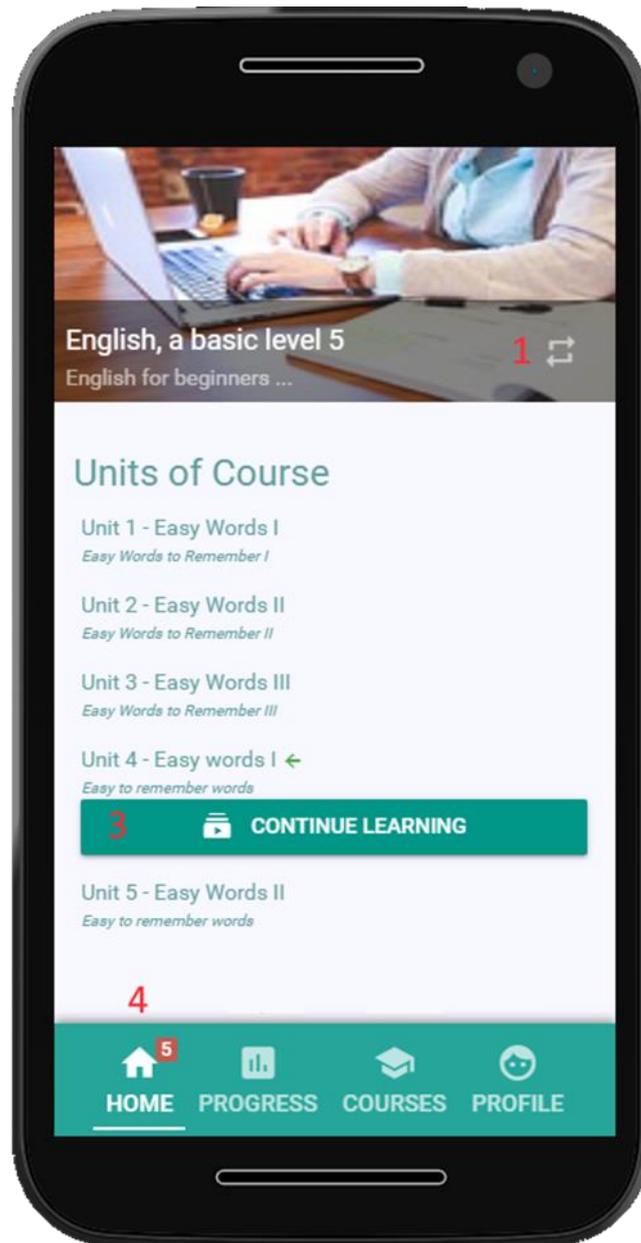


Ilustración 1-5: Modo aprendizaje y modo repaso

Una vez completadas las tarjetas de aprendizaje de una unidad estas serán almacenadas y mediante un algoritmo que determina el momento óptimo de

repaso Lembra nos recordara cuando es el mejor momento para volver a repasarlas. Podemos identificar sí tenemos tarjetas pendientes de repaso y el número de ellas mediante el icono <4>. Para repasar este contenido pulsaremos sobre el botón <1>.

A.3.2 Tarjetas de aprendizaje

En Lembra la unidad mínima de aprendizaje son las tarjetas o también conocidas como *flash cards*. Estas normalmente contienen palabras, imágenes, audio o videos y se pide al usuario que realice cierta interacción con el objetivo de memorizar su contenido.

Existen elementos comunes a todas las tarjetas de aprendizaje independientemente del tipo. Uno de estos es la barra superior <1> y que podemos identificar en la *Ilustración 1-6* e *Ilustración 1-7*, esta permite conocer el avance en el número de tarjetas completadas y restantes. Otro de los elementos comunes en son los botones de avance y retroceso entre las tarjetas disponibles de la unidad <2>. También dispondremos siempre del botón *check* <3> que permite ver la opción correcta de la pregunta en caso de que no la conozcamos.

Actualmente existen dos tipos de tarjetas de aprendizaje y que describimos a continuación.

A.3.2.1 Tarjetas de tipo test

En la *Ilustración 1-6* podemos apreciar una tarjeta de tipo test compuesta por cuatro opciones y siendo solo una la correcta. El objetivo de este tipo de tarjetas es seleccionar la opción correcta de entre las mostradas. En caso de acertar la opción se marcará en color verde. Por el contrario, si la respuesta no es la correcta, esta se marcará en color rojo.



Ilustración 1-6: Tarjeta tipo test

A.3.2.2 Tarjetas de tipo subtítulos de video

En la *Ilustración 1-7* *Ilustración 1-6* podemos apreciar una tarjeta de tipo video, estas tarjetas reproducirán un fragmento de un video de forma automática y nos solicitará se introduzca una de las palabras ocultas del fragmento de video <5>. Mediante el botón <6> podemos ver la traducción de la palabra oculta y mediante el botón <4> podemos repetir el fragmento de video nuevamente.

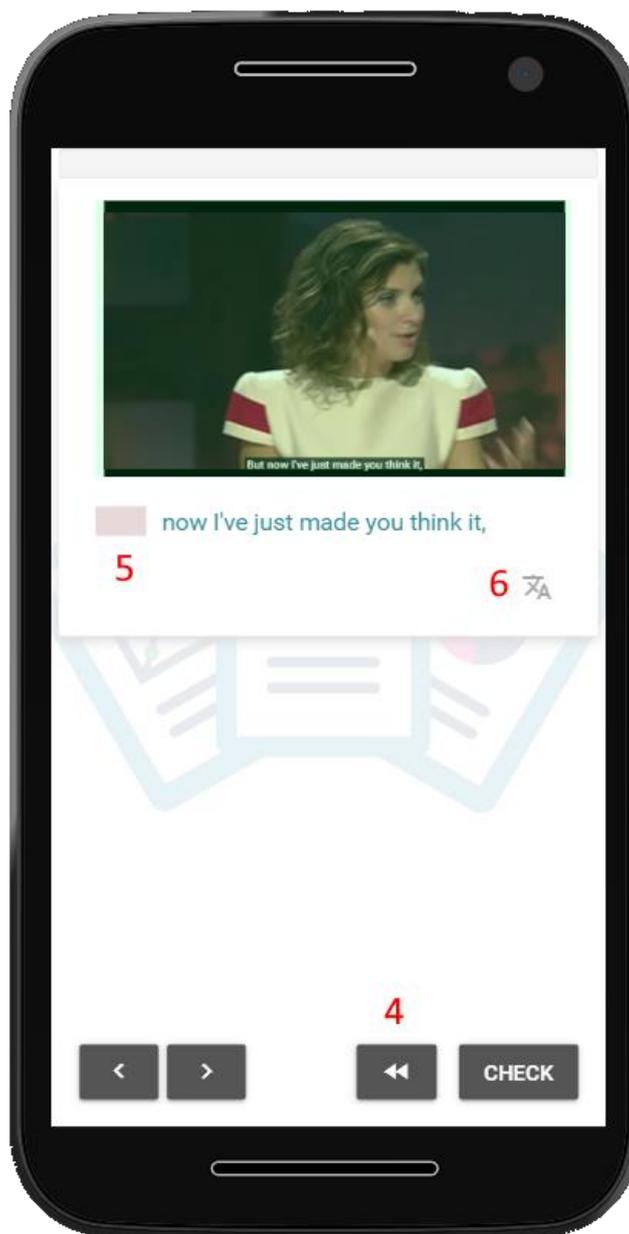


Ilustración 1-7: Tarjeta de tipo video

A.4 Pantalla de Progreso

En esta pantalla podemos visualizar mediante los botones <1> y <2> el progreso diario y semanal del curso que tenemos establecido como curso principal en el momento de la visualización. Podemos ver una representación en la *Ilustración 1-8*.

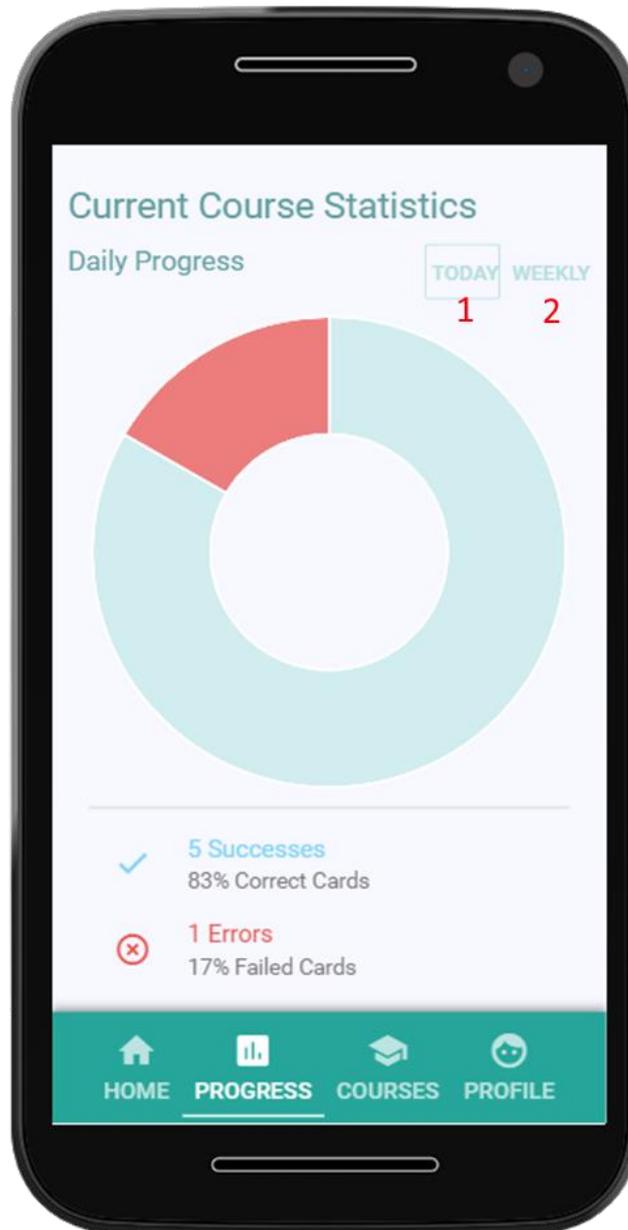


Ilustración 1-8: Pantalla de progreso

A.5 Pantalla de Cursos

A.5.1 Elección de cursos matriculados y principal

En esta opción de menú se visualizarán todos los cursos del sistema en los que podemos matricularnos. Podemos ver esta pantalla en la *Ilustración 1-9*. En caso de no estar matriculado en un curso nos aparecerá la opción *Enrol in course* <2> la cual nos permitirá matricularnos. En caso de que ya estemos matriculados en el curso nos aparecerá el botón *Set Main Course* <1>, que permitirá establecerlo como curso principal, lo cual nos permitirá cursar su contenido.

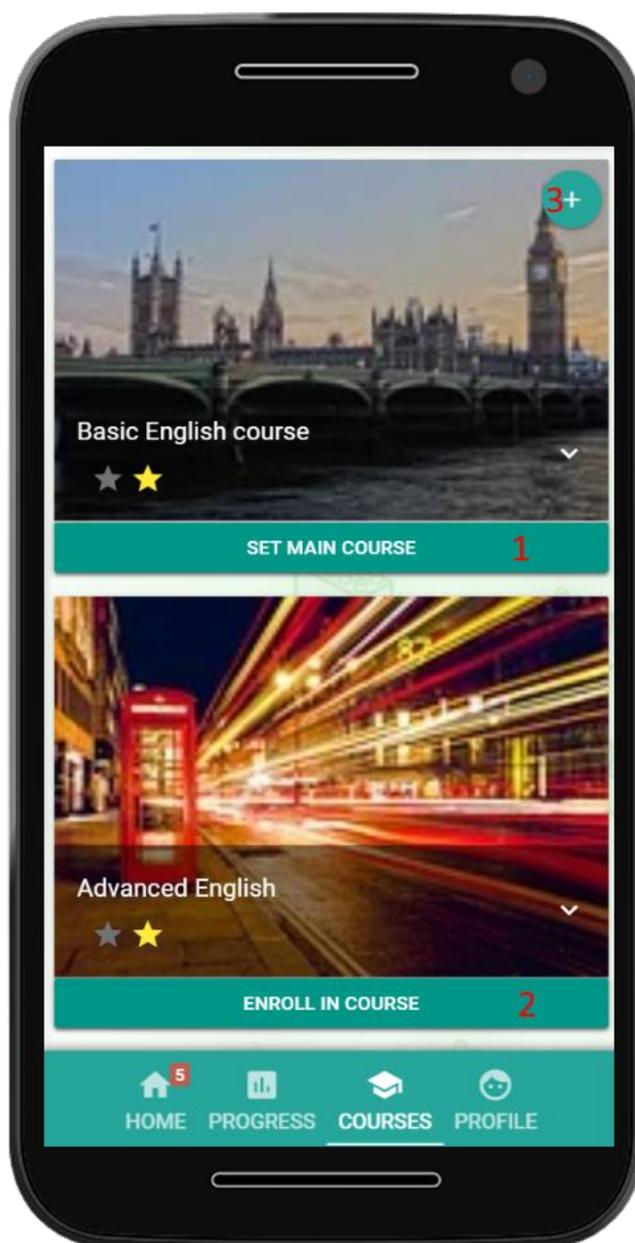


Ilustración 1-9: Pantalla de cursos

A.5.2 Crear un nuevo curso

La creación de cursos está disponible para cualquier usuario autenticado en la aplicación. Mediante el botón <3> representado en la *Ilustración 1-9* obtendremos acceso al asistente de creación de cursos. En primer lugar, este asistente nos solicitara que establezcamos un nombre <4> y una descripción para el curso <5>. Por último, para avanzar seleccionaremos la opción *next* <6>. Podemos ver esta ventana del asistente en la *Ilustración 1-10*.

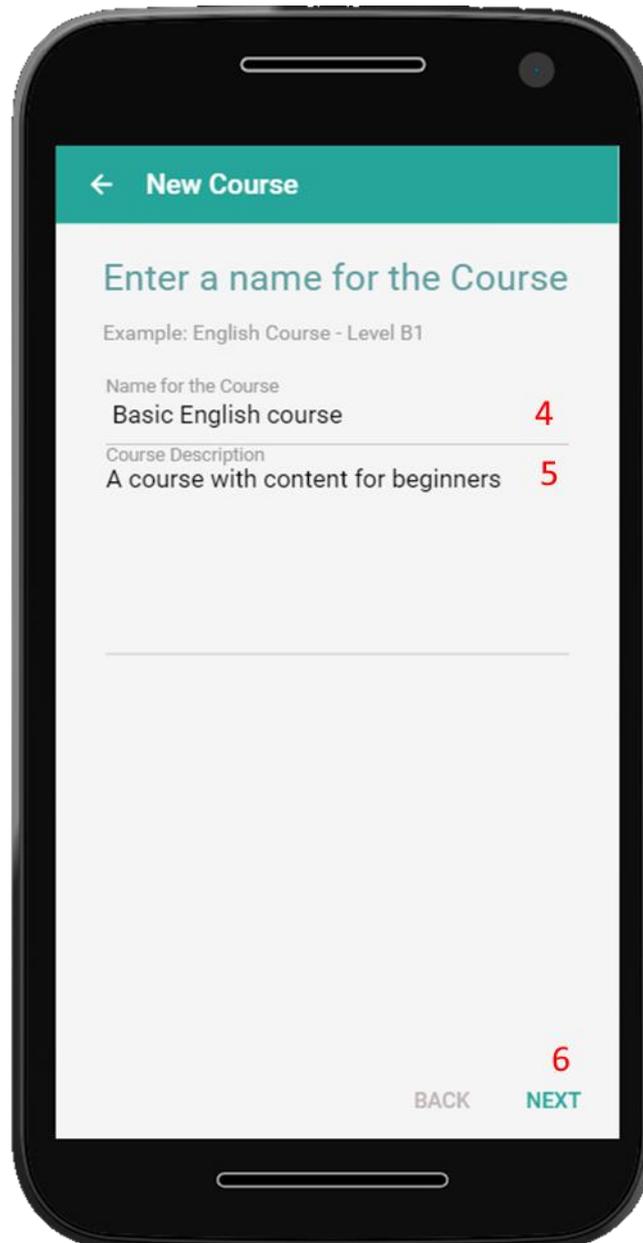


Ilustración 1-10: Asistente nuevo curso I

La siguiente ventana del asistente de creación de un nuevo curso nos solicitara se establezca una imagen principal para el curso. Para esta imagen

puede utilizarse una imagen obtenida de la galería del dispositivo mediante la opción <9> o utilizar el buscador de imágenes integrado en la aplicación <7><8>. Este buscador obtendrá imágenes coincidentes con la palabra de búsqueda. Podemos ver una representación de esto en la *Ilustración 1-11*.

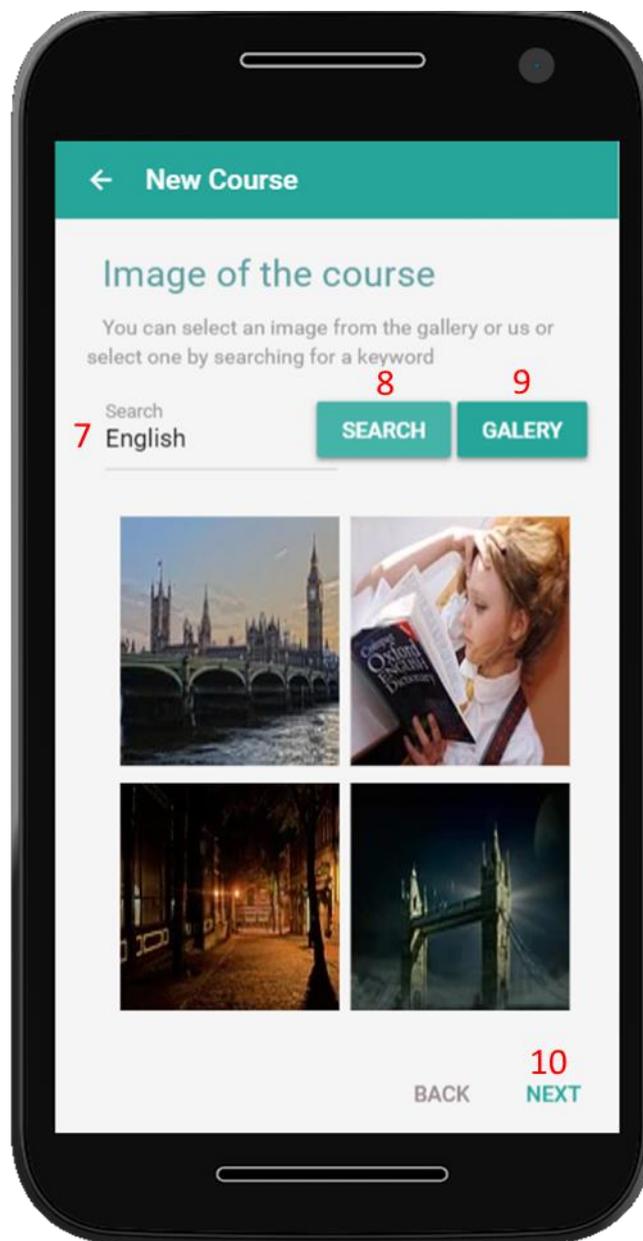


Ilustración 1-11: Asistente nuevo curso II

La siguiente ventana del asistente es la de selección de modalidad del curso. Lembra actualmente permite dos modalidades cursos, cursos de idiomas o cursos para pruebas estandarizadas (tipo test). Podemos apreciar esto en la *Ilustración 1-12*.

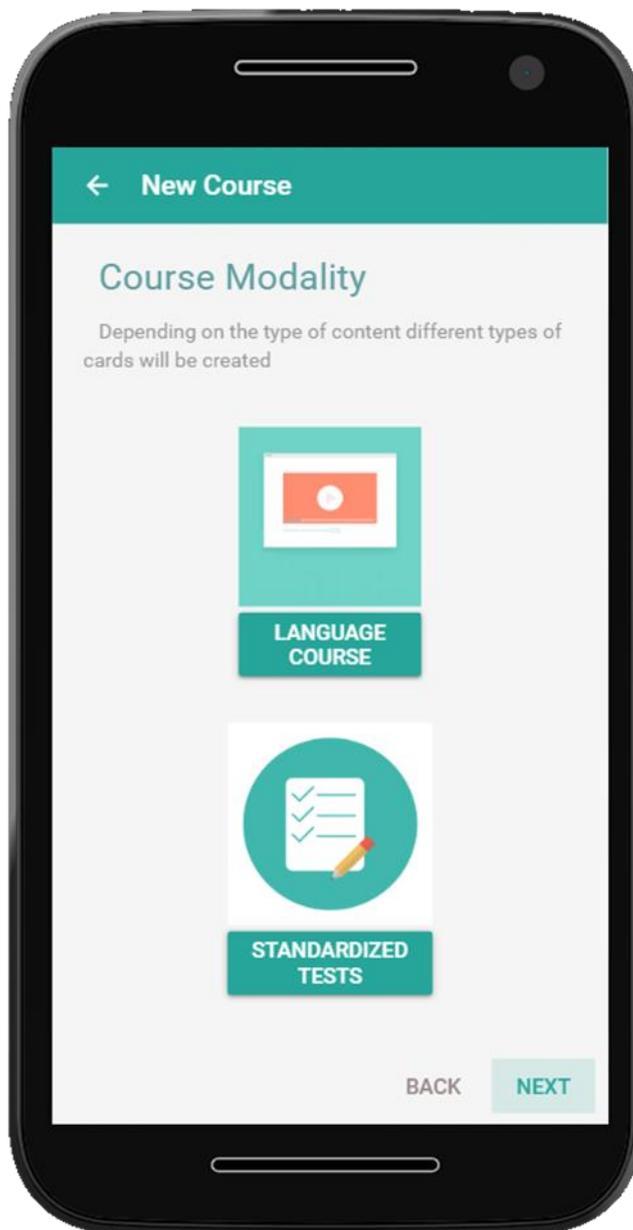


Ilustración 1-12: Asistente nuevo curso III

La última ventana del asistente es la que permite seleccionar la privacidad del curso. En el caso de que establezcamos *check* <11> en público el contenido generado estará visible para todos los usuarios de la aplicación. Podemos ver esta pantalla en la *Ilustración 1-13*.



Ilustración 1-13: Asistente nuevo curso IV

Tras pulsar en el botón <12> se finalizará el asistente de nuevo curso y la aplicación redirigirá a la ventana de edición del curso como podemos apreciar en la *Ilustración 1-14*.

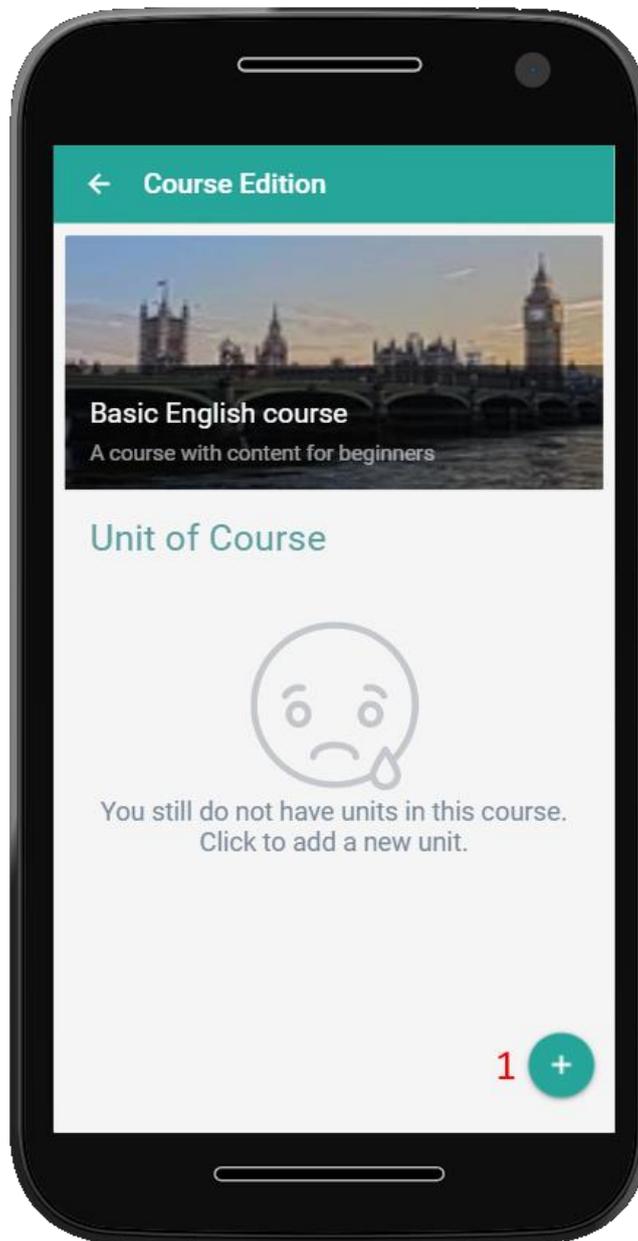


Ilustración 1-14: Edición del curso

Por último, hay que destacar que, de forma inicial, todos los cursos se encuentran deshabilitados, es decir no estarán visibles para el resto de los usuarios de la aplicación. Es importante que estos se habiliten para lo que como mínimo deberán contener al menos una unidad. Podemos ver que cuando el curso cumple este requisito nos aparecer el botón *Enable Course* <7> que podemos apreciar en la *Ilustración 1-16*

A.5.3 Añadir unidades al curso

Cada curso está compuesto por una serie de unidades que a su vez contendrán una serie de tarjetas de aprendizaje. El primer paso cuando creamos un nuevo curso es añadir unidades. Para añadir una unidad accederemos a la pantalla edición de curso que podemos visualizar en la *Ilustración 1-14* mediante el botón <1>. Esto generara una nueva ficha en el contenido del curso como podemos apreciar en la *Ilustración 1-15*.

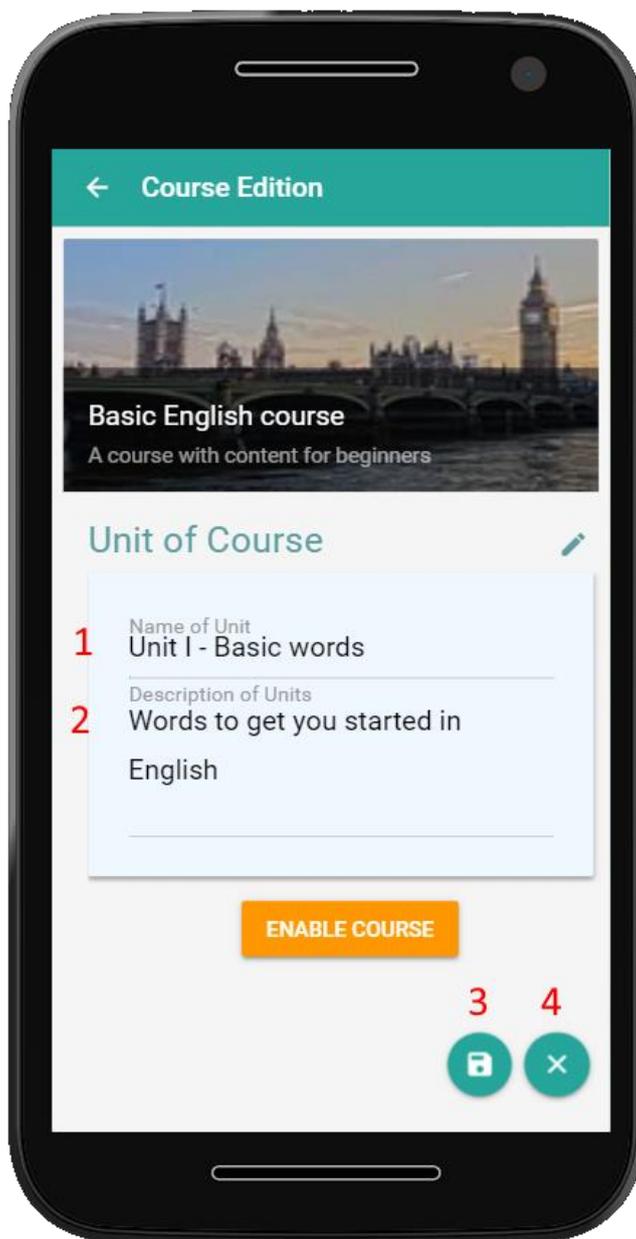


Ilustración 1-15: Nueva unidad I

Para cada nueva unidad se solicitara introducir en los campos de entrada de texto un nombre <1> y una descripción de la unidad <2>, tras completar estos campos procederemos a confirmar o descartar la nueva unidad mediante los botones <3> y <4> respectivamente y que podemos apreciar en la *Ilustración 1-16*. Tras esto, la unidad quedara confirmada y tendremos la posibilidad de modificarla mediante el botón <5> o borrarla mediante el botón <6>.

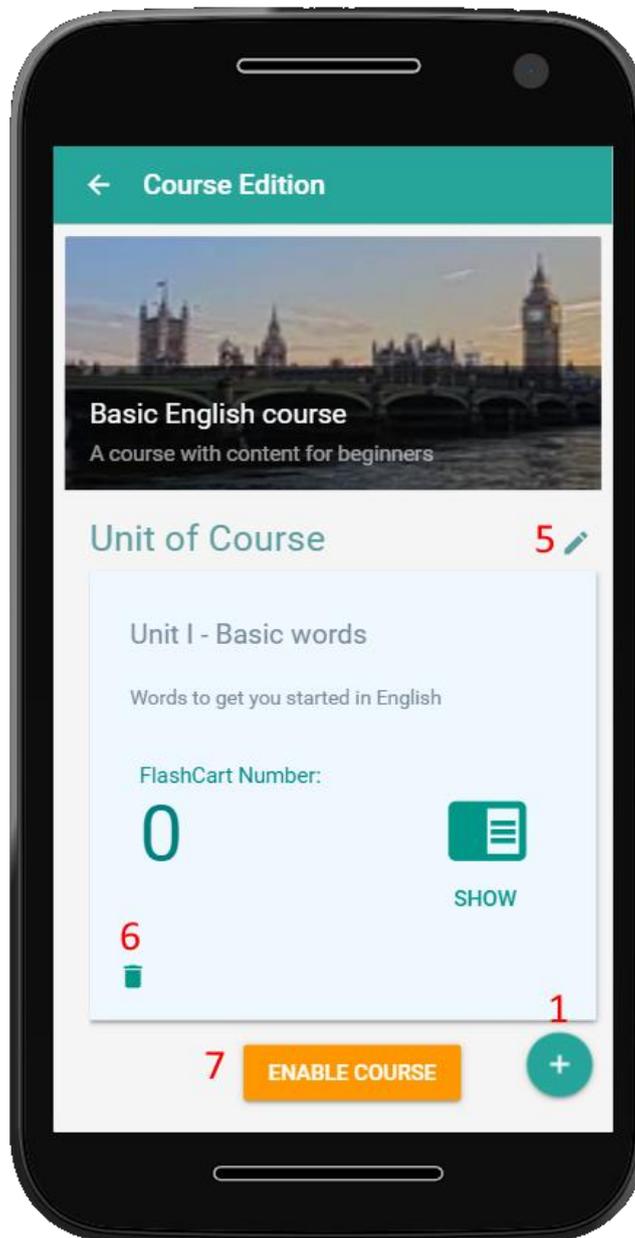


Ilustración 1-16: Nueva unidad II

A.5.4 Añadir tarjetas de aprendizaje al curso

Una tarjeta o *flash card*, es la unidad mínima de aprendizaje en Lembra. Como vimos en la sección 0 existe la posibilidad de crear diferentes tipos de cursos, esto es importante a la hora de crear tarjetas ya que dependiendo del tipo de curso seleccionado podemos crear diferentes tipos de tarjetas de aprendizaje.

Independientemente del tipo de curso, la forma de añadir nuevas tarjetas de aprendizaje para cada unidad siempre será la misma. Podemos añadir nuevas tarjetas de aprendizaje desde la edición de curso seleccionado el botón <1> y que podemos apreciar en la *Ilustración 1-17*.

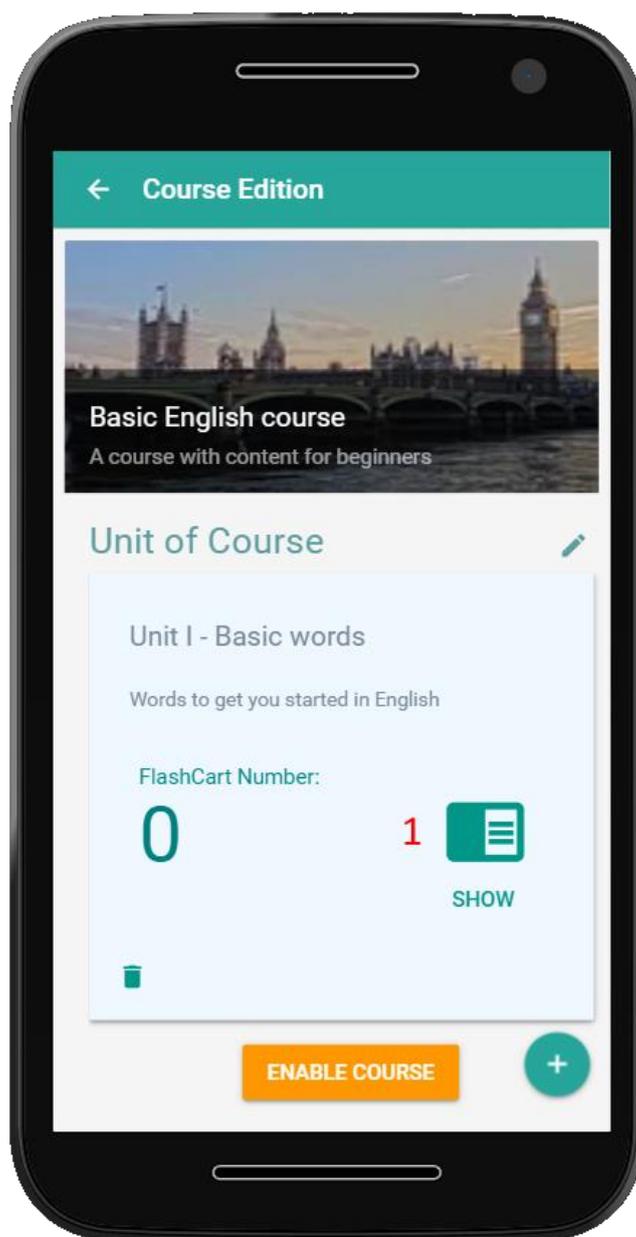


Ilustración 1-17: Nuevas Tarjetas de aprendizaje

A.5.4.1 Tarjetas de aprendizaje de tipo video

Las tarjetas de aprendizaje de tipo video son un tipo de tarjetas de aprendizaje para cursos orientados al aprendizaje de idiomas. Este tipo de tarjetas permiten haciendo uso de los subtítulos de un video, ocultar una cierta palabra para que con posterioridad esta sea recordada por el usuario. Para acceder al asistente de creación de este tipo de tarjetas seleccionaremos el botón <3> que podemos ver en la *Ilustración 1-18*.

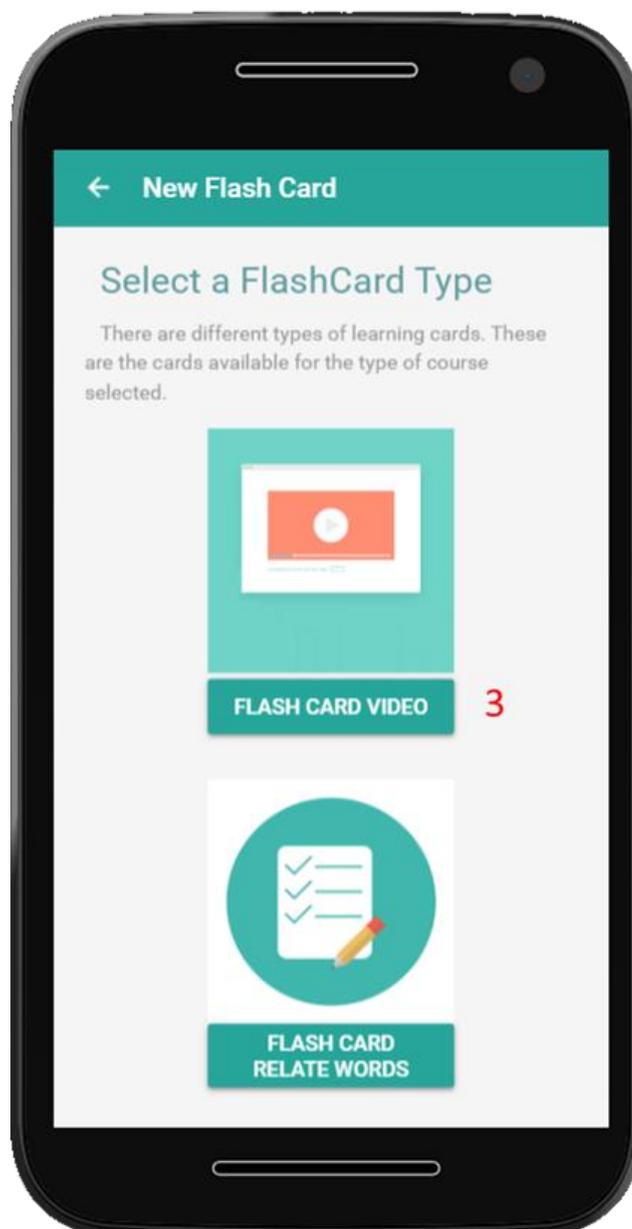


Ilustración 1-18: Tarjetas tipo video

En primer lugar, el asistente nos solicitará una palabra clave mediante el campo de texto <4>, esto permitirá efectuar una búsqueda entre los videos del portal de almacenamiento de videos *YouTube*. Esto se encuentra representado en las ilustraciones: *Ilustración 1-19* e *Ilustración 1-20*

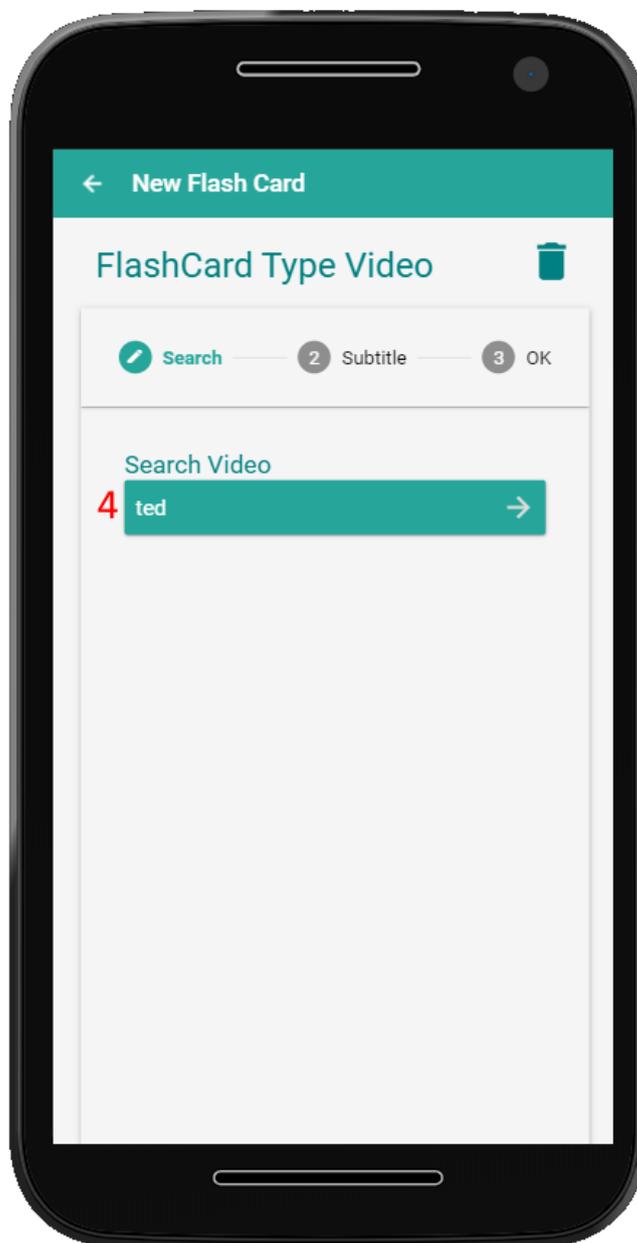


Ilustración 1-19: Asistente nueva tarjeta de video I

En la *Ilustración 1-20* podemos el resultado de búsqueda de las palabras "TED english", podemos editar el criterio de búsqueda mediante el botón <5>. Para seleccionar uno de los videos del resultado de la búsqueda pulsaremos el botón <6> que aparecerá en la parte inferior de cada video.

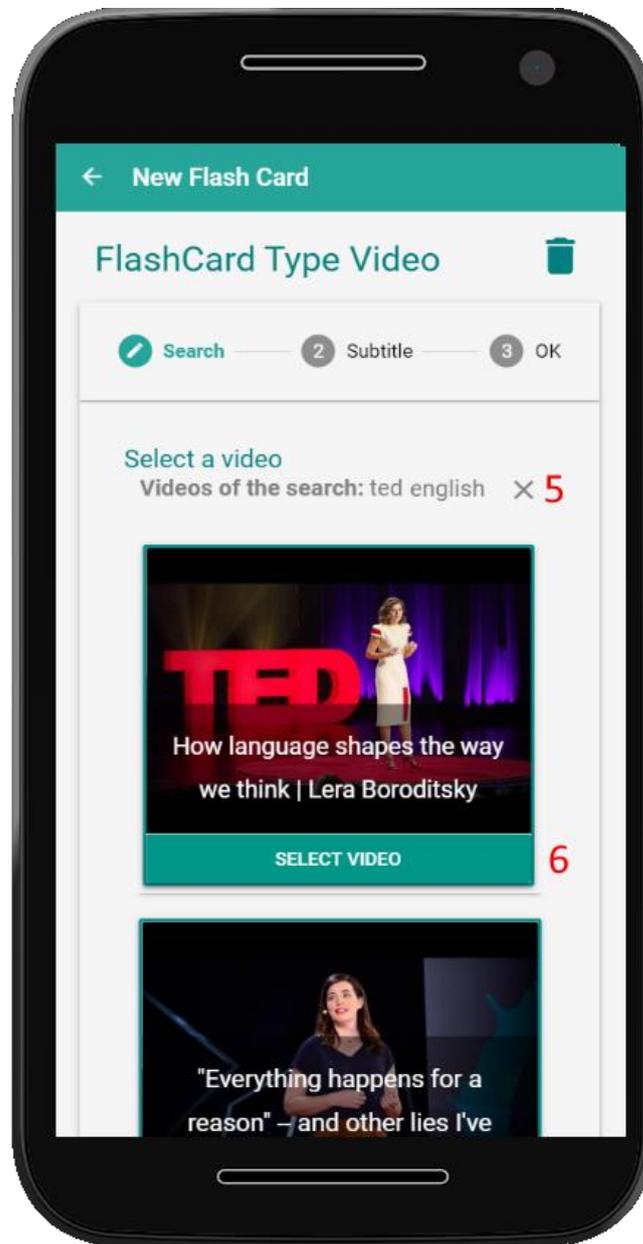


Ilustración 1-20: Asistente nueva tarjeta de video II

Tras seleccionar el video, el asistente nos mostrara todos los subtítulos del video seleccionado divididos en fragmentos. Podemos ver una representación en la *Ilustración 1-21*. Ahora debemos seleccionar un fragmento de subtítulo de los propuestos mediante el botón <7> lo que nos guiara la siguiente ventana del asistente.

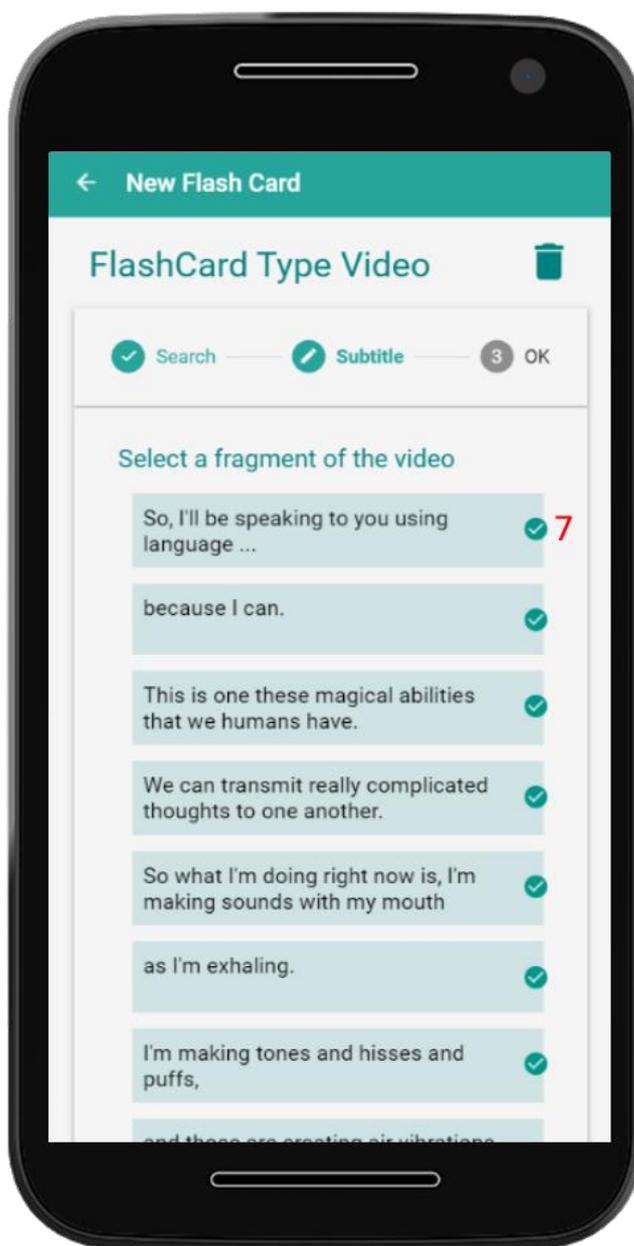


Ilustración 1-21: Asistente nueva tarjeta de video III

En la *Ilustración 1-22* podemos apreciar la última ventana del asistente de creación de la tarjeta de tipo video. Esta muestra en la parte superior el fragmento de video correspondiente con el subtítulo seleccionado. Este fragmento de video es ajustable mediante el botón <8> pudiéndose ajustar el tiempo de inicio y fin del video en caso de que este no se correspondiese con el fragmento del subtítulo mostrado.

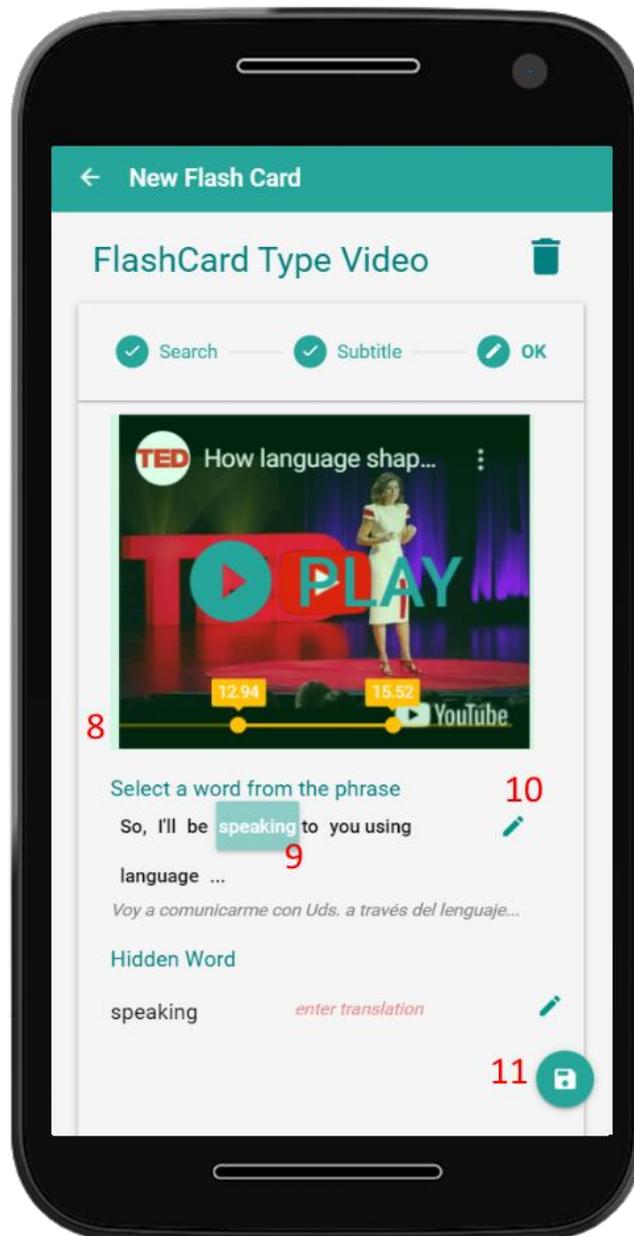


Ilustración 1-22: Asistente nueva tarjeta de video IV

En la parte inferior del video aparecerá el fragmento del subtítulo y del que deberemos seleccionar una palabra <9>, esta será la palabra que el usuario con posterioridad deberá introducir cuando escuche el video. Para finalizar, mediante el botón <11> se guardará la tarjeta de tipo video creada.

Por último, una vez disponemos de tarjetas de aprendizaje en la unidad nos aparecen listadas y podremos borrarlas mediante el botón <13> o añadir nuevas mediante el botón <12>. Podemos ver esto en la *Ilustración 1-23*.

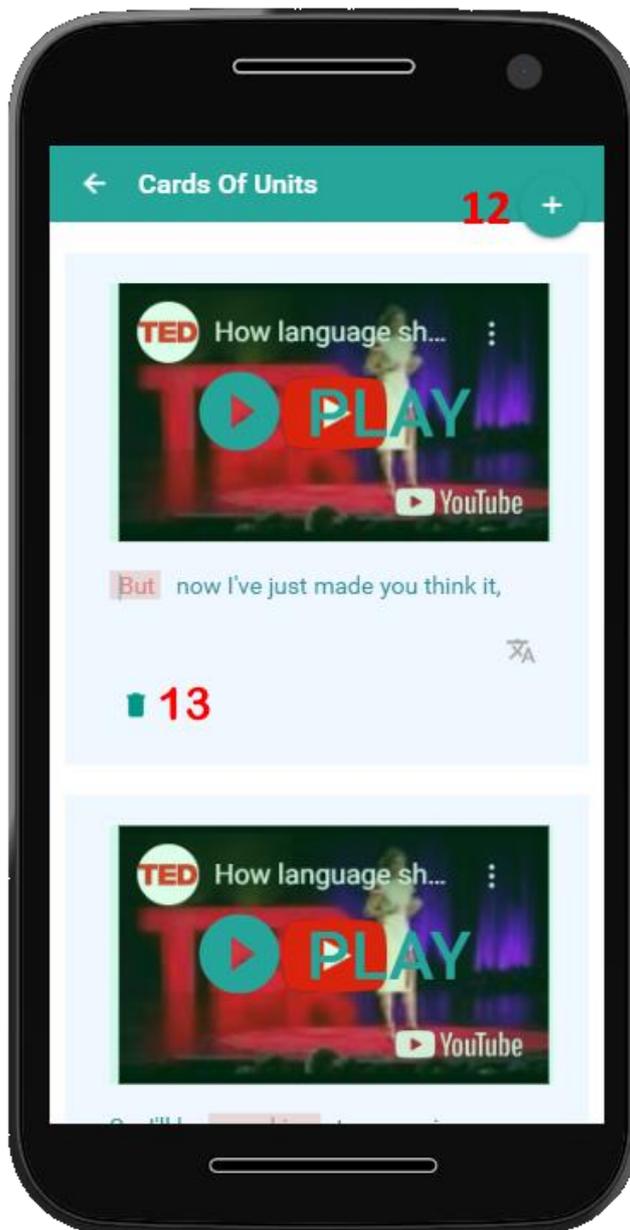


Ilustración 1-23: Listado de tarjetas de unidad

A.6 Pantalla perfil

La pantalla de perfil se encuentra accesible mediante el botón <1> de la barra principal para dispositivos móviles como podemos apreciar en la *Ilustración 1-24*

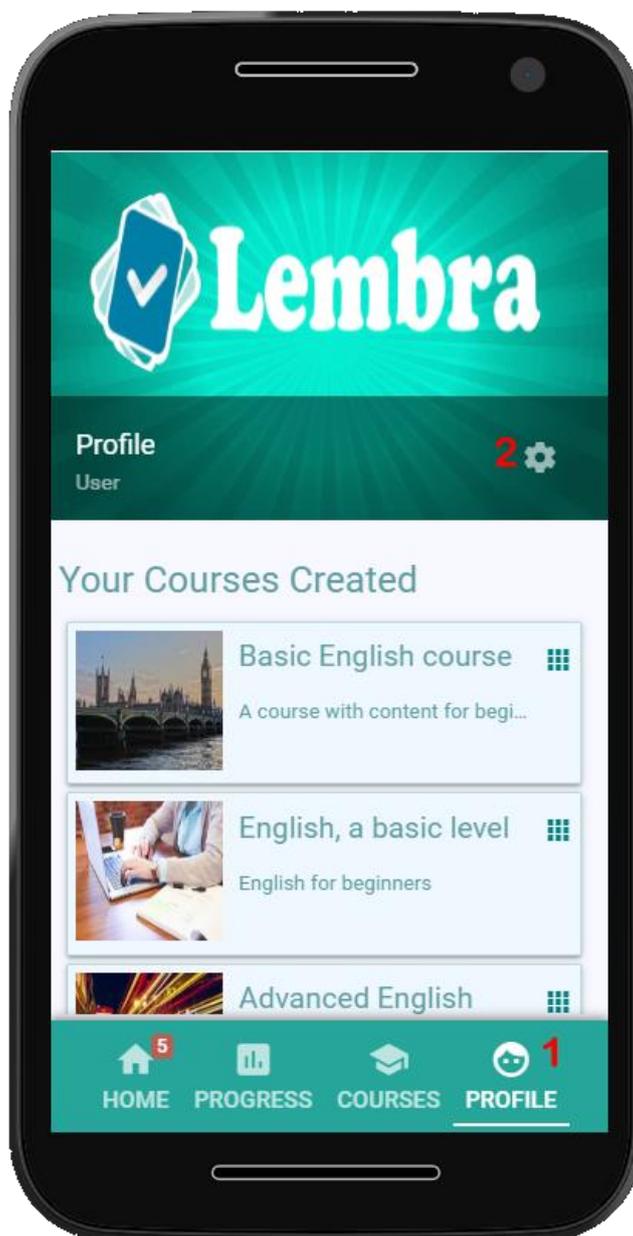


Ilustración 1-24: Pantalla perfil

Esta pantalla nos ofrece el acceso a la configuración del usuario mediante el botón <2> representado en la ilustración. Además, esta pantalla contiene la administración de los cursos creados y matriculados y de lo que hablaremos en las *secciones A.6.1 y A.6.2* de este anexo.

A.6.1 Gestionar cursos creados

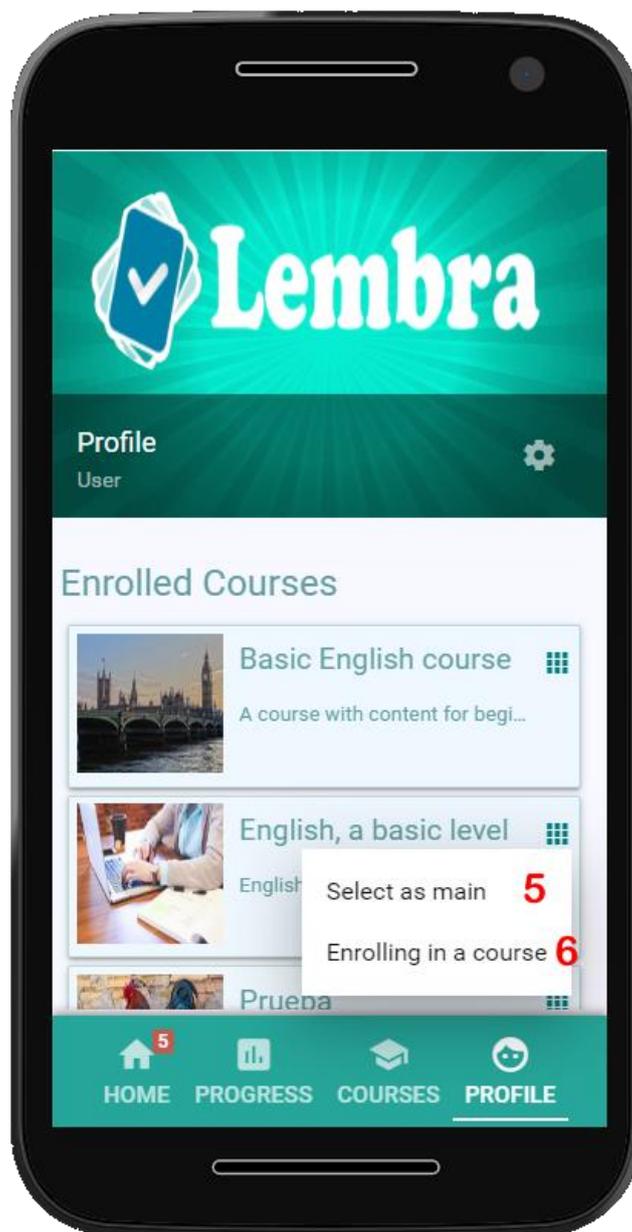
En la sección 0 vimos la posibilidad que ofrece Lembra de crear cursos. En esta pantalla aparecerán todos los cursos creados por el usuario autenticado, podemos apreciar esto en la *Ilustración 1-25*. Mediante la opción <3> y <4> del menú contextual de cada curso podemos borrarlos o acceder la edición que describimos en la *sección A.5.3* de este anexo.



Ilustración 1-25: Pantalla perfil - cursos creados

A.6.2 Gestionar cursos matriculados

Como vimos en la sección 0 Lembra permite estar matriculados en varios cursos simultáneamente e ir alternando el aprendizaje entre los diferentes cursos. Desde la pantalla perfil tendremos acceso al listado de cursos en los que el usuario autenticado se encuentra matriculado pudiendo este gestionar mediante las opciones <5> y <6> del menú contextual de cada curso.



A.6.3 Pantalla de preferencias de usuario.

Como vimos en la *Ilustración 1-24* el acceso se realiza mediante el botón <2>. En esta pantalla se podrán configurar las preferencias de la aplicación. Actualmente algunas de las preferencias que podemos visualizar en la *Ilustración 1-26* no están finalizadas y por lo tanto no tendrán funcionalidad.

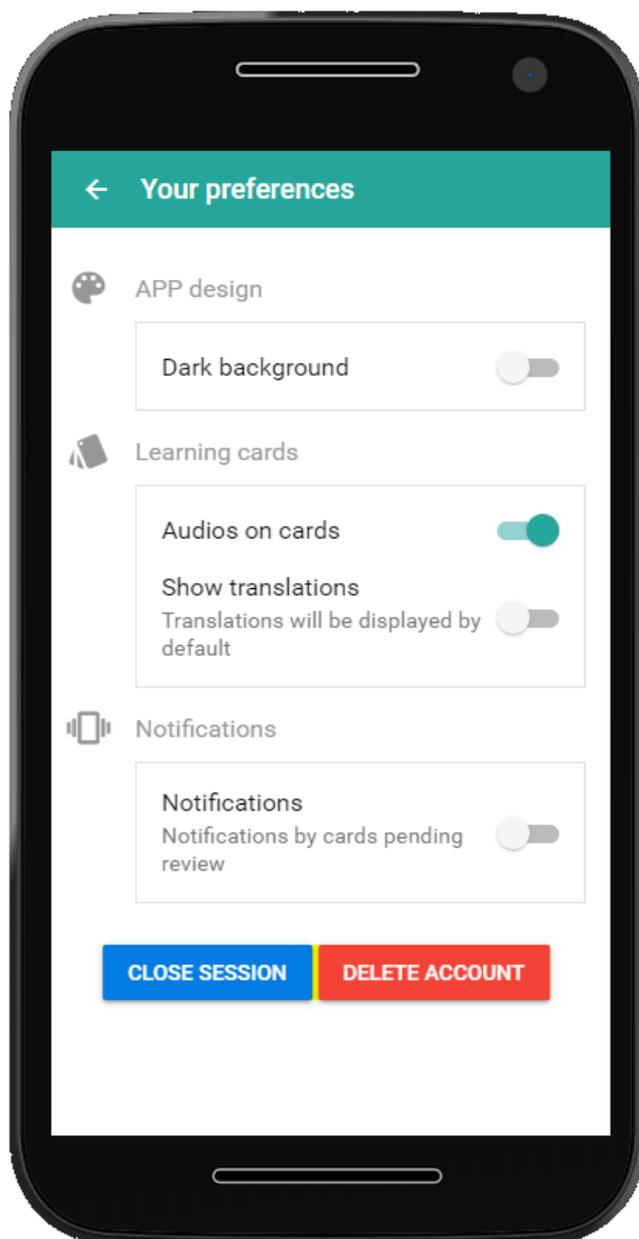


Ilustración 1-26: Preferencias de usuario

Existen diferentes técnicas de aprendizaje, una de estas técnicas de aprendizaje es el repaso espaciado en la que se repasa a intervalos crecientes. Para empezar el sujeto se familiariza con la información para fijarla en su memoria, posteriormente se realizan repasos periódicos y, conforme se va consolidando la información en la memoria, los repasos se distanciarán más y más en el tiempo. Un gran número de investigaciones de la psicología cognitiva y educativa han demostrado que realizar un estudio de forma espaciada en el tiempo es una forma de mejorar la efectividad y la eficiencia del aprendizaje.

En este Trabajo Fin de Grado se desarrolla una aplicación multiplataforma destinada al aprendizaje mediante técnicas de repaso espaciado que favorecen la retención de contenido en la memoria a largo plazo. El dominio en el que se ha puesto en práctica ha sido el aprendizaje de idiomas y las pruebas tipo test para las que han sido diseñados un conjunto de ejercicios los cuales pueden ser creados y editados por los usuarios desde la propia aplicación.

Durante este Trabajo Fin de Grado se abarcará desde el estado de arte, hasta el desarrollo de la aplicación justificando la elección de las tecnologías, la metodología y los métodos de trabajo empleados.

