Universitat Oberta de Catalunya

uoc.edu

Academia virtual de alemán: aprendizaje personalizado con *feedback* correctivo para Android

Jan Bühler Olivé

Máster universitario de desarrollo de aplicaciones para dispositivos móviles

Trabajo final de máster

Francesc D'Assís Giralt Queralt

Carles Garrigues Olivella

30/12/2020

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual 3.0 España de Creative Commons

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Academia virtual de alemán: aprendizaje personalizado con feedback correctivo para Android*
Nombre del autor:	*Jan Bühler Olivé*
Nombre del consultor/a:	*Francesc D'Assís Giralt Queralt*
Nombre del PRA:	*Carles Garrigues Olivella*
Fecha de entrega (mm/aaaa):	12/2020
Titulación:	*Máster universitario de desarrollo de aplicaciones para dispositivos móviles*
Área del Trabajo Final:	*M0.659 – Trabajo final de máster DADM*
Idioma del trabajo:	*Español*
Palabras clave:	*Aprender, alemán y corrección*

Resumen del Trabajo

En este trabajo se presenta el desarrollo de una academia virtual de alemán para la plataforma Android. Concretamente, se trata de una solución mediante la cual el alumnado puede cursar las distintas unidades didácticas que conforman los niveles A1, A2, B1 y B2 de alemán a través de un aprendizaje personalizado, como lo hace una persona nativa y con *feedback* correctivo, que a su vez son las características diferenciadoras respecto a otras apps de idiomas existentes en el mercado.

El trabajo cubre las fases de planificación, análisis, diseño, implementación y pruebas del ciclo de vida de un sistema de información. Destacar la aplicación de la filosofía del diseño centrado en el usuario (DCU), la elaboración de casos de uso y la definición de la arquitectura del sistema en la fase de diseño y el uso del IDE Android Studio, el lenguaje de programación Kotlin y las funcionalidades de *authentication*, r*ealtime database* y *storage* de la plataforma en la nube Firebase de Google en la fase de implementación.

Finalmente, remarcar que la aplicación desarrollada en este trabajo se puede adaptar a la enseñanza de prácticamente cualquier otra temática, como las matemáticas, la geografía, la historia, la preparación de oposiciones, la teoría del carnet de conducir, etc. para las cuales también puede ser muy útil la presentación de contenidos de forma organizada y fomentar la interacción entre alumnado y profesorado. En cualquier caso, la clave está en la generación de contenidos adecuados y de calidad.

Abstract

This work presents the development of a virtual german academy for the Android platform. Specifically, it's a solution by which students can take the different didactic units that make up the A1, A2, B1 and B2 of german levels through personalized learning, as a native person does and with corrective feedback, which in turn, are the differentiating characteristics respectly to other language apps on the market.

The work covers the planning, analysis, design, implementation and testing phases of the life cycle of an information system. Highlight the application of the philosophy of user-centered design (UCD), the elaboration of use cases and the definition of the system architecture in the design phase and the use of the Android Studio IDE, the Kotlin programming language and the authentication, realtime database and storage functionalities of Google's Firebase cloud platform in the implementation phase.

Finally, note that the application developed in this work can be adapted to the teaching of practically any other subject, such as mathematics, geography, history, preparation for examinations, driving license theory, etc. Since, for all of them, it can also be very useful to present content in an organized way and to encourage the interaction between students and teachers. In any case, the key is generate adequate and quality content.

Índice

Índice.....iii
Índice de figuras.....vi
Índice de tablas.....viii
1. Introducción.....1
1.1 Contexto y justificación del trabajo.....1
1.1.1 Estudio de mercado.....3
1.2 Objetivos del trabajo.....8
1.3 Enfoque y método seguido.....10
1.3.1 Tecnología seleccionada.....10
1.3.2 Lenguaje de programación: Java o Kotlin.....11
1.3.3 Modelo de desarrollo.....12
1.4 Planificación del trabajo.....14
1.4.1 Entorno de desarrollo.....14
1.4.2 Planificación.....15
1.4.3 Cálculo del coste en el mercado.....19
1.4.4 Riesgos y medidas de contingencia.....19
1.5 Breve sumario de productos obtenidos.....21
2. Diseño centrado en el usuario (DCU).....22
2.1 Usuarios y contexto de uso.....22
2.1.1 Análisis de la competencia.....22
2.1.2 Entrevista.....22
2.2 Diseño conceptual.....24
2.2.1 Perfiles de usuario.....24
2.2.2 Personas y escenarios.....24
2.3 Prototipado.....28
2.3.1 Sketching.....28
2.3.2 Árbol de navegación.....30
2.3.3 Prototipo de alta fidelidad.....31
2.4 Evaluación.....34
2.4.1 Evaluación del árbol de navegación.....34

2.4.2 Evaluación del prototipo de alta fidelidad....34
3. Diseño técnico de la aplicación....37
3.1 Definición de los casos de uso....37
3.2 Diseño de la arquitectura....44
3.2.1 Diagrama de clases....44
3.2.2 Patrón de arquitectura de software....45
4. Implementación....47
4.1 Inicialización y configuración del proyecto....47
4.1.1 Creación del proyecto....47
4.1.2 Integración de Firebase en el proyecto....47
4.1.3 Configuración de Firebase Authentication....47
4.1.4 Creación de la Firebase Realtime Database....47
4.1.5 Inicialización de Firebase Cloud Storage....48
4.1.6 Librería RxJava/RxAndroid....48
4.1.7 Librería GlideToVectorForYou....48
4.1.8 Librería PhotoView....49
4.1.9 Librería Material Drawer....49
4.2 Transformación del diagrama de clases en una estructura de nodos JSON....50
4.2.1 Desnormalización del esquema JSON....51
4.2.2 Integración de la estructura de nodos JSON en Firebase Realtime Database....52
4.3 Utilización de la aplicación sin conexión a Internet....52
4.3.1 Persistencia en disco de datos de Firebase Realtime Database....52
4.3.2 Persistencia en disco y sincronización de contenidos multimedia mediante Firebase Cloud Storage....53
4.3.4 Consumo de espacio en disco de la aplicación....54
4.4 Descarga de contenidos....55
4.5 Utilización de gráficos vectoriales escalables (SVG)....56
4.6 Estructura del proyecto....57
4.6.1 Reutilización: layout includes, layouts de componentes, adapter parametrizado y fragmentos....58
4.6.2 Comunicación entre la actividad Page y los fragmentos de tipo de ejercicio....59

4.7 Envío de ejercicios mediante correo electrónico para obtener *feedback* correctivo....59
4.8 Animaciones de paginación de un libro....59
4.9 Sección de créditos....60
4.10 Herramientas Lint y Android Profiler....60
5. Pruebas....61
5.1 Creación de tests unitarios....61
5.2 Pruebas funcionales....62
5.3 Pruebas por parte de personas usuarias colaboradoras....62
6. Evaluación de la evolución del trabajo....63
6.1 Hito: entrega de la PEC2 – Diseño....63
6.2 Hito: entrega de la PEC3 – Implementación....63
6.3 Hito: entrega de la PEC4 – Entrega final....64
7. Conclusiones....66
8. Glosario....69
9. Bibliografía....70
10. Anexos....75
10.1 Instrucciones de instalación, configuración, compilación y ejecución de la aplicación....75
10.2 Manual de usuario de la aplicación....77
10.2.1 Introducción....77
10.2.2 Registro....77
10.2.3 Autenticación....78
10.2.4 Selección de nivel de alemán....78
10.2.5 Selección de unidad didáctica....79
10.2.6 Navegación entre páginas y al índice de unidades didácticas....79
10.2.7 Realización de ejercicios....80
10.2.8 Consulta de créditos y cerrar la sesión....80

Índice de figuras

Figura 1: Conocimiento del alemán en la Unión Europea (Qué idioma, 2020).....2
Figura 2: Comparativa de características de las apps analizadas.....6
Figura 3: Modelo de desarrollo en cascada.....12
Figura 4: Distribución de las versiones de Android (XDA Developers, 2020).....15
Figura 5: Diagrama de Gantt de la PEC1.....17
Figura 6: Diagrama de Gantt de la PEC2.....17
Figura 7: Diagrama de Gantt de la PEC3.....18
Figura 8: Diagrama de Gantt de la PEC4.....18
Figura 9: Sketching 1 de 2.....28
Figura 10: Sketching 2 de 2.....29
Figura 11: Árbol de navegación.....30
Figura 12: Iniciar sesión - Registro - Selección de nivel de alemán.....31
Figura 13: Unidades didácticas del nivel A1 - El alfabeto I - El alfabeto II.....31
Figura 14: Los artículos I - Los artículos II - Los artículos III.....32
Figura 15: Unidades didácticas del nivel A2 - Presentarse I - Presentarse II.....32
Figura 16: Unidades didácticas del nivel B1 - Comprensión lectora.....33
Figura 17: Unidades didácticas del nivel B2 - Expresión escrita - Comprensión oral.....33
Figura 18: Acceso al índice de unidades didácticas desde cualquier pantalla de una unidad didáctica específica.....34
Figura 19: Botón de menú hamburguesa.....34
Figura 20: Prototipo inicial: español como idioma preferente.....35
Figura 21: Prototipo final: alemán como idioma preferente.....35
Figura 22: En el nivel B2 ya no consta traducción en español del enunciado.....35
Figura 23: Prototipo inicial, colores azul y blanco para todos los niveles e indicación de nivel en todas las pantallas.....35
Figura 24: Prototipo final, uso de un color distinto para cada nivel y sin indicación de nivel en todas las pantallas.....35
Figura 25: Uso de un color distinto para cada nivel.....36
Figura 26: Prototipo inicial, botones en la parte inferior de la pantalla.....36
Figura 27: Prototipo final, botones en la barra de tareas.....36
Figura 28: Diagrama de los casos de uso.....37
Figura 29: Arquitectura del sistema.....44
Figura 30: Diagrama de clases UML.....45
Figura 31: Arquitectura MVP.....46
Figura 32: Establecer orientación vertical.....47
Figura 33: Estructura de nodos JSON.....50

Figura 34: Estructura de nodos JSON desnormalizada.....51
Figura 35: Estructura de directorios para organizar los contenidos multimedia.....53
Figura 36: Sincronización de recursos multimedia de una unidad didáctica que requiere recursos.....54
Figura 37: Estructura del proyecto en Android Studio.....57
Figura 38: Interfaz PageListener.....59
Figura 39: Menú de acceso a la sección de créditos.....60
Figura 40: Sección de créditos.....60
Figura 41: Tests unitarios.....61
Figura 42: Android Studio Rebuild Project.....75
Figura 43: Extracto de dependencias del archivo build.gradle.....76
Figura 44: Seleccionar configuración y dispositivo, y ejecutar.....76
Figura 45: Opción "Regístrate".....77
Figura 46: Pantalla de registro.....77
Figura 47: Pantalla de autenticación.....78
Figura 48: Pantalla de selección de nivel de alemán.....78
Figura 49: Pantalla de selección de unidad didáctica.....79
Figura 50: Barra de navegación.....79
Figura 51: Enunciado del ejercicio.....80
Figura 52: Menú de la aplicación.....80
Figura 53: Pantalla de créditos.....80

Índice de tablas

Tabla 1: Análisis de la app uTalk - Aprenda cualquier idioma....3
Tabla 2: Análisis de la app Aprender alemán....3
Tabla 3: Análisis de la app Duolingo....4
Tabla 4: Análisis de la app Aprende idiomas gratis - Mondly....4
Tabla 5: Análisis de la app Babbel....5
Tabla 6: Análisis de la app Busuu....5
Tabla 7: Análisis de la app LearnMatch: aprender idiomas, aprender inglés....6
Tabla 8: Objetivos principales del trabajo....8
Tabla 9: Requisitos funcionales del trabajo....8
Tabla 10: Requisitos no funcionales del trabajo....9
Tabla 11: Objetivos opcionales del trabajo....9
Tabla 12: Ventajas e inconvenientes de los lenguajes de programación Java y Kotlin....11
Tabla 13: Relación entre las PECs y las fases del modelo de desarrollo en cascada....13
Tabla 14: Horas de dedicación al trabajo....15
Tabla 15: Planificación, costo y estado de los hitos del proyecto....16
Tabla 16: Caso de uso CU01. Registrar persona usuaria....38
Tabla 17: Caso de uso CU02. Autenticar persona usuaria....38
Tabla 18: Caso de uso CU03. Seleccionar nivel de alemán....39
Tabla 19: Caso de uso CU04. Seleccionar unidad didáctica....39
Tabla 20: Caso de uso CU05. Realizar una página de teoría....40
Tabla 21: Caso de uso CU06. Realizar un ejercicio del tipo escuchar opciones estáticas....40
Tabla 22: Caso de uso CU07. Realizar un ejercicio del tipo escuchar opciones aleatorias....41
Tabla 23: Caso de uso CU08. Realizar un ejercicio del tipo seleccionar la opción correcta....41
Tabla 24: Caso de uso CU09. Realizar un ejercicio del tipo expresión escrita....42
Tabla 25: Caso de uso CU10. Realizar un ejercicio del tipo comprensión lectora....42
Tabla 26: Caso de uso CU11. Realizar un ejercicio del tipo comprensión oral....43
Tabla 27: Caso de uso CU12. Enviar ejercicio para su corrección....43

1. Introducción

1.1 Contexto y justificación del trabajo

El trabajo final de máster que se ha decidido desarrollar es una **academia virtual de alemán** en forma de app. Hoy día vivimos en un mundo global que nos permite trabajar prácticamente en cualquier parte del mundo, por ejemplo, se puede vivir en España y trabajar para una empresa estadounidense, alemana, francesa, etc. Del mismo modo, la globalización también ha provocado el incremento de la demanda (Travel Student, 2020) y en consecuencia de las posibilidades de aprender una lengua extranjera y, una de las más solicitadas actualmente es mediante una app (MSUToday, 2020), principalmente porque permite aprender en cualquier momento, en cualquier lugar y durante el tiempo de que se disponga.

La popularidad de las apps en general, es debida a que actualmente gran parte de los habitantes del mundo poseen un teléfono inteligente y, además, es el dispositivo con el que interaccionan mayoritariamente con Internet (Review 42, 2020), principalmente porque siempre lo llevan encima y por su facilidad de uso. Si a esto le sumamos la importante bajada de las tarifas de datos y el despliegue de la tecnología 5G las posibilidades son infinitas.

Las personas usuarias de apps para aprender idiomas, además de querer aprender rápido, en cualquier sitio y en cualquier momento, quieren aprender como lo hace una persona nativa y poder realizar un seguimiento de su progreso. Y este es precisamente el objetivo principal de la app presentada aquí: **ofrecer un aprendizaje personalizado, como lo hace una persona nativa y con *feedback* correctivo** que, a día de hoy, sigue siendo uno de los métodos más eficaces de aprender, ya que es fundamental conocer cuáles son y dónde se comenten los errores. Concretamente, la app actúa de interfaz entre el alumnado y el profesorado nativo, que se encarga de ajustar el aprendizaje al nivel de cada estudiante a través de unidades didácticas personalizadas, con el objetivo de transmitirle todo el conocimiento necesario para lograr que en poco tiempo sea capaz de construir oraciones y sepa utilizarlas en pequeñas conversaciones.

A partir de cierto nivel, el alumnado tendrá la opción de redactar textos de distintos tipos y temática, y de acceder a textos de temas específicos que deberán leer, resumir y sobre los cuales tendrán que responder a preguntas. Todo ello será corregido por el profesorado nativo y el alumnado recibirá el correspondiente *feedback* correctivo. Por tanto, se produce una interacción entre el profesorado y el alumnado que posibilita ajustar su aprendizaje con el objetivo de corregir rápidamente sus errores.

Otra finalidad de la app es la preparación del alumnado para los exámenes oficiales de idiomas, por lo que tendrán la opción de practicar todos los tipos de ejercicios que aparecen en este tipo de exámenes: gramática, vocabulario, comprensión oral, comprensión lectora, expresión escrita, etc. El mismo autor

de este trabajo habría sido usuario de esta app de haber existido para el idioma catalán cuando se preparaba para los exámenes oficiales de los distintos niveles. Al no existir y tener la necesidad de interaccionar con profesorado nativo, especialmente para la corrección de errores en la expresión escrita y para la preparación de las pruebas, se vio obligado a apuntarse a unos cursos presenciales, a los que tenía que acudir a horas intempestivas, y donde para asistir a una hora de curso necesitaba invertir casi tres. Esto resulta muy complicado para la mayoría de la gente trabajadora con niños y otras responsabilidades.

Inicialmente, el idioma ofrecido es el alemán a causa de los conocimientos del mismo por parte del autor de este trabajo, por su relación con personas que se dedican a la enseñanza de este idioma, por su popularidad como segundo, tercer o cuarto idioma en muchos países y por su gran oferta laboral y educativa. Además, es la lengua materna más hablada en la Unión Europea –por 97 millones de personas nativas y 170 millones de personas en total en Europa (Qué idioma, 2020)– aunque la principal razón de su creciente demanda, es la estabilidad económica de los países donde es oficial: Alemania, Austria, Suiza, Luxemburgo, Bélgica y Liechtenstein.

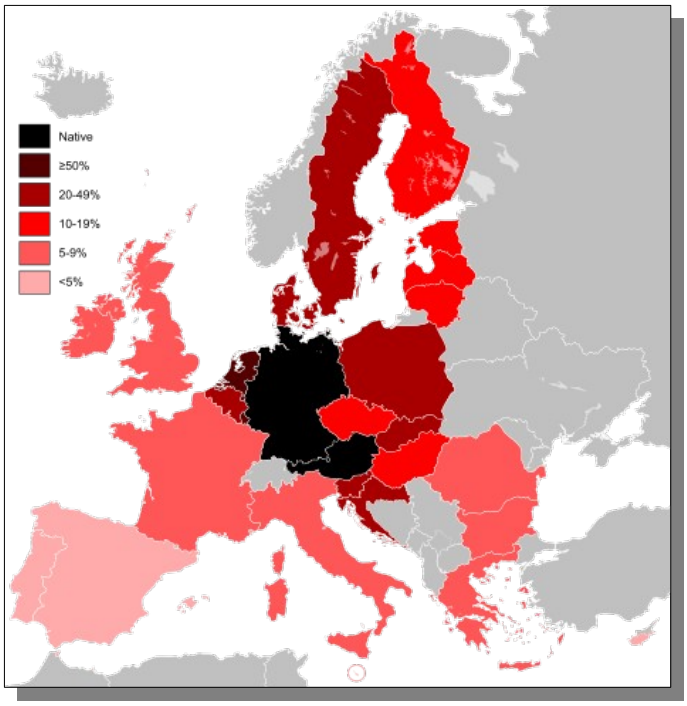

Figura 1: Conocimiento del alemán en la Unión Europea (Qué idioma, 2020)

Por otro lado, en España, también se debe tener en cuenta la demanda de aprendizaje de alemán por motivos laborales, especialmente por quienes se dedican a la industria del turismo.

En cualquier caso, a largo plazo se pueden incorporar otros idiomas siempre y cuando se consideren sus peculiaridades y se cuente con la participación del profesorado nativo requerido. También a largo plazo, se podría estudiar la

posibilidad de integrar el uso de la app presentada aquí en las escuelas, tendiendo en cuenta que una de sus principales ventajas es la generación continua de material personalizado al alumnado y sus errores.

Finalmente, comentar que la pandemia de coronavirus (COVID-19) está provocando la cancelación de gran cantidad de cursos de idiomas, por lo que toda alternativa de poder seguir aprendiendo es una oportunidad.

1.1.1 Estudio de mercado

A continuación se presenta un pequeño análisis de diversas apps para aprender idiomas, entre ellos el alemán:

<table>
<tr><th colspan="2">uTalk – Aprenda cualquier idioma</th></tr>
<tr><td colspan="2">

App para aprender vocabulario de más de 140 idiomas a través pequeños y sencillos juegos.
Disponible para Android y para iOS (uTalk, 2020).
Se ofrece una versión demo.
Para quienes se registran, se ofrece la posibilidad de acceder a temas específicos mediante compras in-app.
Productos de compra integrados en aplicaciones: entre 1,69 € y 59,99 € por elemento.

</td></tr>
<tr><th>Puntos fuertes</th><th>Puntos débiles</th></tr>
<tr><td>Juegos como por ejemplo el Memory, donde se combinan palabras, con imágenes y sonido.</td><td>La versión demo pronto se vuelve monótona y repetitiva.</td></tr>
<tr><td>Aprendizaje de vocabulario a través de imágenes y su pronunciación a través del audio.</td><td>No se proporciona feedback de pronunciación y uso adecuado de las palabras.</td></tr>
<tr><td>Permite practicar la comprensión oral.</td><td></td></tr>
</table>

Tabla 1: Análisis de la app uTalk - Aprenda cualquier idioma

<table>
<tr><th colspan="2">Aprender alemán</th></tr>
<tr><td colspan="2">

App consistente en una colección de listas de vocabulario de diferentes temas: ir de compras, los colores, los números, etc.
Disponible para Android y para iOS (Aprender alemán, 2020).
Productos de compra integrados en aplicaciones: 169 € por elemento.

</td></tr>
<tr><th>Puntos fuertes</th><th>Puntos débiles</th></tr>
<tr><td>App gratuita.</td><td>La pronunciación en alemán deja mucho que desear.</td></tr>
<tr><td>Permite escuchar las palabras en español y en alemán.</td><td>No existen actividades para practicar el vocabulario aprendido.</td></tr>
<tr><td>Las listas de vocabulario están en español y en alemán.</td><td>No hay feedback correctivo de ningún tipo.</td></tr>
</table>

Tabla 2: Análisis de la app Aprender alemán

Duolingo	
• App para aprender inglés, francés, italiano, alemán y portugués. • Disponible para Android y para iOS (Duolingo, 2020). • Para tener acceso a todas las funcionalidades es necesario registrarse y pagar. • Productos de compra integrados en aplicaciones: entre 1,09 € y 134,39 € por elemento.	
Puntos fuertes	Puntos débiles
Versión gratuita con gran cantidad de actividades	Incordio continuo de la invitación para pasar a Duolingo Plus.
Ajuste automático del nivel de dificultad al nivel del usuario.	No existe interacción de ningún tipo con profesorado o personas nativas.
Se puede utilizar sin conexión.	El *feedback* correctivo de limita a la presentación de la solución de las actividades por parte de la app.
Es posible crear una cuenta de usuario que posibilita el registro de las puntuaciones obtenidas.	

Tabla 3: Análisis de la app Duolingo

Aprende idiomas gratis - Mondly	
• App que ofrece la posibilidad de aprender 33 idiomas. • Disponible para Android y para iOS (Mondly, 2020). • Para tener acceso a todas las funcionalidades es necesario registrarse y pagar. • Productos de compra integrados en aplicaciones: entre 4,99 € y 191,99 € por elemento.	
Puntos fuertes	Puntos débiles
Aprendizaje de vocabulario a través de su relación con imágenes.	Se han detectado errores en actividades de gramática y traducciones incorrectas.
La app presenta el progreso del alumno en forma de puntos y estadísticas como la cantidad de vocabulario aprendido en un determinado tiempo.	No queda claro qué funcionalidades requieren registro y pago.
Ranking de puntuación a nivel de grupo de amigos, país y mundial.	Algunos usuarios se quejan de que la app se bloquea.
Diálogos en forma de chat de diferentes situaciones como, por ejemplo, la visita a un restaurante.	El *feedback* correctivo de limita a la presentación de la solución de las actividades por parte de la app.

Tabla 4: Análisis de la app Aprende idiomas gratis - Mondly

Babbel	
• App y plataforma web de cursos online. • Disponible para Android y para iOS (Babbel, 2020). • La versión gratuita permite realizar algunas lecciones. • Productos de compra integrados en aplicaciones: entre 9,95 € y 79,99 € por elemento.	
Puntos fuertes	Puntos débiles
Aprendizaje de 14 idiomas con actividades de vocabulario, gramática, comprensión lectora, comprensión oral y expresión oral (a través de reconocimiento de voz) creadas por un equipo didáctico.	El reconocimiento de voz no siempre funciona correctamente, lo cuál impide el aprendizaje y provoca malas puntuaciones.
La persona usuaria puede crear su propio plan personal de actividades.	El *feedback* correctivo es solo a través de la misma app, no a través de profesorado.
Se puede utilizar sin conexión.	

Tabla 5: Análisis de la app Babbel

Busuu	
• Red social y plataforma interactiva para aprender idiomas. • Disponible para Android y para iOS (Busuu, 2020). • Productos de compra integrados en aplicaciones: entre 0,50 € y 149,99 € por elemento.	
Puntos fuertes	Puntos débiles
Aprendizaje de 12 idiomas a través de más de 80 lecciones para cada uno de ellos, con actividades de vocabulario, gramática, comprensión lectora, comprensión oral y expresión oral.	No hay control ni garantía sobre los conocimientos de las personas que conforman la comunidad, lo cuál puede provocar un aprendizaje incorrecto.
Comunidad de 100 millones de usuarios.	Peligro que supone la comunicación descontrolada entre por ejemplo de niños o adolescentes con otras personas.
Entrenador de vocabulario mediante el uso de inteligencia artificial.	Quejas sobre los precios.
Pruebas de nivel.	Incordio continuo de la invitación para abonarse.
Conversaciones con hablantes nativos.	
Se puede utilizar sin conexión.	
Feedback correctivo de actividades de expresión escrita y de expresión oral, por parte de personas nativas.	
Posibilidad de crear su propio plan de estudio.	
Presentación gráfica del progreso del estudiante.	
Cursos de viaje para aprender lo imprescindible para viajar.	
Cursos para empresas	

Tabla 6: Análisis de la app Busuu

LearnMatch: aprender idiomas, aprender inglés	
• App que ofrece el aprendizaje de seis idiomas: inglés, español, francés, portugués, italiano y alemán. • Disponible para Android y para iOS (LearnMatch, 2020). • Productos de compra integrados en aplicaciones: entre 2,29 € y 54,99 € por elemento.	
Puntos fuertes	Puntos débiles
Función que permite invitar a sus amigos para competir contra ellos en una partida en tiempo real.	No existen actividades de gramática.
Aprendizaje de vocabulario a través de imágenes.	Se aprende vocabulario pero no en qué situaciones y cómo debe utilizarse.
Funcionalidad que permite crear su propio diccionario, aunque requiere el registro y pago.	En la versión en alemán, las palabras se presentan sin artículo, por lo que quien pretende aprender ya empieza mal desde el principio y más teniendo en cuenta que en alemán hay tres artículos.
	El *feedback* correctivo de limita a la presentación de la solución de las actividades por parte de la app.

Tabla 7: Análisis de la app LearnMatch: aprender idiomas, aprender inglés

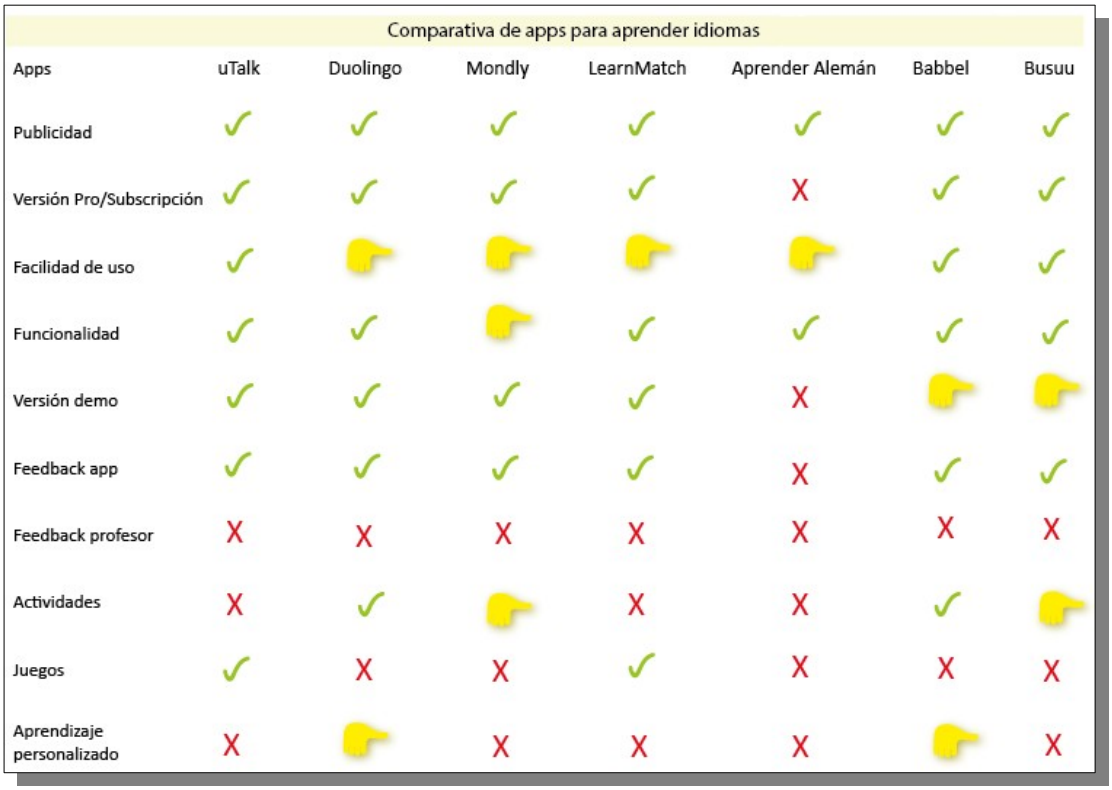

Figura 2: Comparativa de características de las apps analizadas

Durante el proceso de investigación realizado, se ha observado que todas las apps de idiomas analizadas tienen publicidad, que todas excepto *Aprender Alemán* ofrecen una versión demo para presentar su servicio a las personas usuarias y que en las de *Duolingo* y *Busuu* se incordia continuamente a la persona usuaria con una invitación para pasar a la versión de pago.

En general, se ha visto que muchas de las apps para aprender alemán existentes se limitan al aprendizaje de vocabulario, en algunos casos mediante juegos como el *Memory*, lo que está bien para empezar pero solo permite avanzar hasta cierto punto: las personas usuarias aprenden un conjunto de palabras y expresiones pero no adquieren conocimientos sobre cómo construir oraciones con ellas ni en qué situaciones utilizarlas. ¿Qué ocurre con las personas usuarias de estas apps? Se acaban aburriendo o incluso frustrando y de un momento a otro pierden el interés en la app y, en el peor de los casos, abandonan por completo el proyecto *aprender una lengua extranjera*. La mayoría de apps de este tipo no dan respuesta a cuestiones como ¿qué necesita la o el estudiante? ¿qué más puede aprender? ¿dónde están sus debilidades? Cuestiones a las cuales se pretende dar respuesta mediante la app presentada en este trabajo.

Por otro lado, existen plataformas como *Busuu*, que ofrecen un servicio de tándem o red social, es decir, ponen a disposición de las personas usuarias una plataforma que permite que se enseñen mutuamente. Lamentablemente, esta solución no siempre acaba cumpliendo con las expectativas de quienes participan, que en muchos casos desisten en su uso porque, a causa de sus limitaciones en el lenguaje expresivo en la lengua extranjera que están aprendiendo, no son capaces o se ven superados al intentar interaccionar con personas nativas. Además, existen participantes que no dominan su propia lengua como para enseñar o corregir a quienes aprenden y consecuentemente el aprendizaje resulta incorrecto. Los dos problemas expuestos en este párrafo no se producen si se decide aprender mediante la app presentada en este trabajo.

En cuanto a las actividades a realizar en las apps analizadas, son genéricas, es decir, no se ajustan a las debilidades y errores de quien pretende aprender (aprendizaje personalizado). Además, del *feedback* correctivo se encarga la misma app, no profesorado nativo, por lo que el alumnado se encuentra solo durante todo el proceso de aprendizaje, con un *feedback* pobre y sin ayuda. En estas condiciones, ¿cómo se puede saber si se ha comprendido correctamente la materia?

1.2 Objetivos del trabajo

En este apartado, se presentan los objetivos principales, los requisitos funcionales, los requisitos no funcionales, y los objetivos opcionales del trabajo. Respecto, a los objetivos opcionales, solamente se llevarán a cabo en caso de producirse un remanente de tiempo cuando se hayan logrado los objetivos principales y se hayan cumplido los requisitos funcionales.

Objetivos principales	
OP01	Crear una app que implemente una academia virtual de alemán.
OP02	Ofrecer un conjunto de unidades didácticas de los cursos de los niveles A1, A2, B1 y B2 de alemán.
OP03	Proporcionar un método de aprendizaje como si se tratase de una persona nativa.
OP04	Preparar al alumnado para los exámenes oficiales de alemán de los niveles A1, A2, B1 y B2.
OP05	Implementación de un sistema de registro y autenticación de usuarios.
OP06	Para las personas registradas posibilitar el aprendizaje personalizado mediante la interacción con el profesorado, principalmente a través de *feedback* correctivo.
OP07	Facilidad de uso de la app a través de una interfaz de usuario simple e intuitiva.
OP08	Llevar a cabo el análisis, la planificación, el diseño y la implementación de una app.
OP09	Aplicar los conocimientos adquiridos en el máster.

Tabla 8: Objetivos principales del trabajo

Requisitos funcionales	
RF01	Registro de una nueva cuenta de usuario.
RF02	Autenticación de un usuario existente.
RF03	Acceder a las unidades didácticas del curso del nivel A1 de alemán.
RF04	Acceder a las unidades didácticas del curso del nivel A2 de alemán.
RF05	Acceder a las unidades didácticas del curso del nivel B1 de alemán.
RF06	Acceder a las unidades didácticas del curso del nivel B2 de alemán.
RF07	Presentación de la parte teórica de una unidad didáctica.
RF08	Resolución de los ejercicios de una unidad didáctica.
RF09	Enviar un ejercicio para su corrección.
RF10	Enviar un documento de texto para su corrección.

Tabla 9: Requisitos funcionales del trabajo

Requisitos no funcionales	
RNF01	El registro y la autenticación de usuarios se llevará a cabo mediante una plataforma móvil en la nube.
RNF02	Las unidades didácticas serán almacenadas en una plataforma móvil en la nube.
RNF03	La aplicación se podrá utilizar sin conexión.
RNF04	La interacción entre alumnado y profesorado se realizará mediante mensajes de correo electrónico.

Tabla 10: Requisitos no funcionales del trabajo

Objetivos opcionales	
OO01	Incluir una prueba de nivel.
OO02	Incluir un modo demo con una unidad didáctica de cada nivel.
OO03	Evaluación del alumnado mediante tests periódicos.
OO04	Incluir un sistema de monetización a través de la inclusión de publicidad, un modelo de suscripción y compras *In-App*.
OO05	Incorporar un sistema de puntos con un *ranking* de usuarios por nivel y con premios, como por ejemplo una corrección gratis.
OO06	Envío de notificaciones push con información sobre nuevas unidades didácticas, premios, etc.
OO07	Adición de más idiomas base como el inglés, francés, etc.

Tabla 11: Objetivos opcionales del trabajo

1.3 Enfoque y método seguido

El trabajo presentado aquí se corresponde al desarrollo de un producto nuevo, concretamente de una academia virtual de alemán en forma de app.

1.3.1 Tecnología seleccionada

Para implementar la app se ha decidido utilizar la tecnología nativa Android por los siguientes motivos:

1. Es una de las tecnologías más utilizadas actualmente tanto a nivel mundial (Statcounter, 2020) como a nivel nacional (Statista, 2020).
2. No tiene limitaciones de acceso a las funcionalidades del dispositivo, de rendimiento y de experiencia de usuario.
3. Ofrece una mayor velocidad de ejecución y respuesta.
4. Permite la publicación de la app en App Stores.
5. Ofrece funcionalidades de monetización de la app.
6. Permite utilizar la app sin conexión.
7. Ofrece un mayor nivel de seguridad.

Comentar que también se ha evaluado la posibilidad de implementar la app como Web App o como App Híbrida pero estas opciones se han descartado porque:

1. Tienen limitaciones de acceso a las funcionalidades del dispositivo, de rendimiento y de experiencia de usuario. El proyecto requiere el acceso a las funcionalidades del dispositivo en su versión inicial y, especialmente, en sus versiones futuras donde podría ser necesario el uso de la cámara, la geolocalización, etc.
2. Presentan limitaciones en cuando a seguridad, funcionalidad sin conexión y velocidad de ejecución y respuesta, que podrían ser problemáticas para la implementación de la app.

En las Web Apps, además de presentar las limitaciones anteriores, no es posible la publicación en App Stores, lo cuál supone un problema para el proyecto presentado aquí porque se trata del principal canal de difusión que se pretende utilizar.

Respecto al desarrollo nativo Android, lógicamente también tiene desventajas:

1. Requiere mayor inversión en desarrollo: se asume porque el estudiante ha orientado el máster al conocimiento de esta tecnología.

2. Código no reutilizable: se asume y, en caso de éxito de la app sobre la plataforma Android, se crearía la correspondiente versión para iOS.

3. Según el tipo de actualizaciones, deben realizarse por iniciativa del usuario: se intentarán minimizar con soluciones como la distribución de las unidades didácticas a través de una plataforma móvil en la nube.

1.3.2 Lenguaje de programación: Java o Kotlin

Actualmente, para el desarrollo de aplicaciones nativas Android, es posible utilizar los lenguajes de programación Java y Kotlin.

A continuación las ventajas e inconvenientes de estos lenguajes:

Java	
Ventajas	Inconvenientes
Respaldado por Oracle.	Código extenso (verbosidad).
Lenguaje muy extendido y muy versátil.	No proporciona seguridad sobre los nulos.
Gran comunidad de soporte.	Soporte a medias con la API de Android: por ejemplo con las funciones lambda.
Es multiplataforma	Requiere mucha memoria.
Kotlin	
Ventajas	Inconvenientes
Respaldado por JetBrains, Google y Spring Framework. Está completamente integrado en el IDE Android Studio.	Al principio cuesta un poco leer y entender el código.
Programación funcional.	Comunidad de soporte pequeña.
Codificación corta, concisa y directa.	Tamaño de los APK ligeramente más grandes.
Proporciona seguridad sobre los nulos.	Compilación más lenta que en Java
100% interoperable con Java.	
Es multiplataforma.	

Tabla 12: Ventajas e inconvenientes de los lenguajes de programación Java y Kotlin

Para la implementación de este proyecto se ha decidido utilizar el lenguaje de programación Kotlin, ya que, además de las ventajas descritas en la tabla anterior:

1. Apunta a ser el lenguaje de programación del futuro para el desarrollo de aplicaciones móviles en Android. Un motivo de peso son las disputas de

Google con la propietaria de Java, Oracle. Uno de los objetivos prioritarios de Google es que Android deje de depender de Java.

2. Google está haciendo esfuerzos importantes, como por ejemplo, la creación junto a JetBrains (empresa creadora de Kotlin) de la Kotlin Foundation para desarrollar el lenguaje codo con codo. Además, Kotlin se está utilizando en desarrollos importantes, como Amazon Web Services, Netflix, Prezi, Pinterest, etc.

1.3.3 Modelo de desarrollo

Como se trata de un proyecto en el cuál los requisitos son conocidos, están fijados y no van a cambiar durante el ciclo de vida del desarrollo se ha optado por utilizar el modelo de desarrollo en cascada (*waterfall*). Este modelo dota de estructura al proceso de desarrollo mediante una clara división en fases que son ejecutadas de forma secuencial y donde los resultados de cada fase son trasladados a la siguiente fase.

Las fases del modelo de desarrollo en cascada son:

1. Análisis: especificación de los requisitos.
2. Diseño: definición de la arquitectura de la aplicación.
3. Implementación: codificación en un lenguaje de programación.
4. Verificación: comprobación del correcto funcionamiento.
5. Mantenimiento: puesta en servicio del software y mantenimiento continuo.

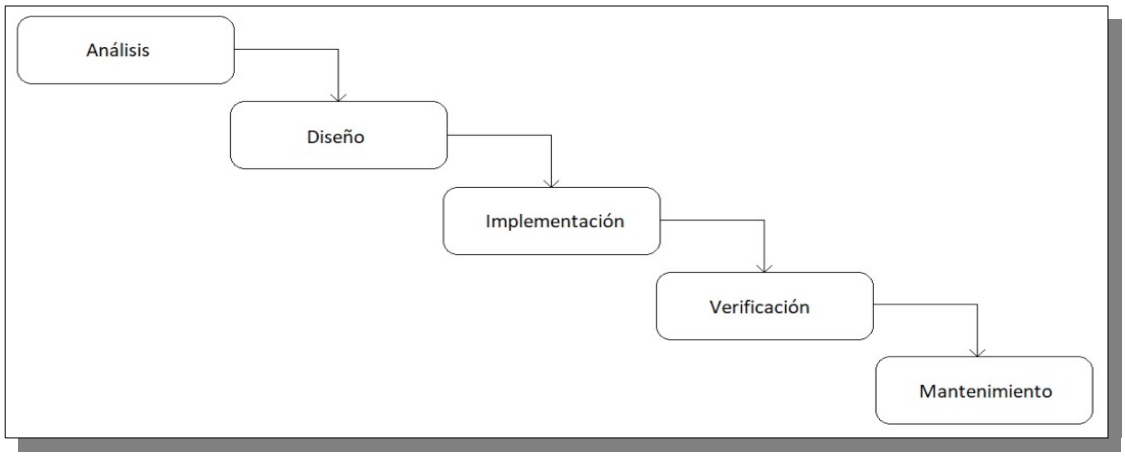

Figura 3: Modelo de desarrollo en cascada

Finalmente, comentar que cada una de las prácticas de evaluación continua (PECs) que conforman este trabajo, casa muy bien con una de las fases del modelo de desarrollo en cascada:

Práctica de evaluación continua	Fase del modelo de desarrollo en cascada
PEC 1	Análisis
PEC 2	Diseño
PEC 3	Implementación
PEC 4	Verificación

Tabla 13: Relación entre las PECs y las fases del modelo de desarrollo en cascada

1.4 Planificación del trabajo

1.4.1 Entorno de desarrollo

Actualmente, para el desarrollo de apps nativas Android se utiliza el entorno de desarrollo integrado (IDE) Android Studio. En este caso se utilizará la última versión estable, que a día de hoy es la 4.0.1 y que se ha instalado en un portátil MacBook Pro con procesador Dual-Core Intel Core i5 de 2,3GHz, 8GB de RAM, con tarjeta gráfica Intel Iris Plus Graphics 640 y sistema operativo macOS Catalina 10.15.7.

Por otro lado, para verificar el correcto funcionamiento del desarrollo se cuenta con los siguientes dispositivos móviles:

1. Teléfono inteligente Motorola Moto G edición Google Play con 1GB de RAM, 8GB de almacenamiento, procesador Quad-Core a 1,2GHz y versión de Android 4.4.4.

2. Tablet Samsung Galaxy Tab S2 SM-T819 con 3GB de RAM, 32GB de almacenamiento, procesador Octa-Core a 1.8GHz y versión de Android 7.

3. Teléfono inteligente Samsung Galaxy M21 con 4GB de RAM, 64GB de almacenamiento, procesador Octa-Core Exynos 9611 2.3GHz y versión de Android 10.

Destacar que el hecho de disponer de dispositivos físicos con distintas características *hardware* y con diferentes versiones de Android, resultará muy beneficioso para la realización de pruebas y la verificación del correcto funcionamiento de la app. El objetivo es que se pueda utilizar sobre el mayor número posible de dispositivos, por lo que se pretende establecer la versión 4.4 de Android, como la versión mínima requerida. Como se puede observar en la siguiente tabla de distribución de las versiones de Android (XDA Developers, 2020), es la menor versión que, con un 4%, se utiliza en un porcentaje de dispositivos que se puede considerar significativo.

Android Platform Version (API Level)	Distribution (as of April 10, 2020)
Android 4.0 "Ice Cream Sandwich" (15)	0.2%
Android 4.1 "Jelly Bean" (16)	0.6%
Android 4.2 "Jelly Bean" (17)	0.8%
Android 4.3 "Jelly Bean" (18)	0.3%
Android 4.4 "KitKat" (19)	4%
Android 5.0 "Lollipop" (21)	1.8%
Android 5.1 "Lollipop" (22)	7.4%
Android 6.0 "Marshmallow" (23)	11.2%
Android 7.0 "Nougat" (24)	7.5%
Android 7.1 "Nougat" (25)	5.4%
Android 8.0 "Oreo" (26)	7.3%
Android 8.1 "Oreo" (27)	14%
Android 9 "Pie" (28)	31.3%
Android 10 (29)	8.2%

Figura 4: Distribución de las versiones de Android (XDA Developers, 2020)

Finalmente, comentar que se hará uso de la plataforma móvil en la nube para el desarrollo de aplicaciones web y móvil Firebase de Google (Firebase, 2020). Inicialmente se utilizará la funcionalidad de *authentication* para la autenticación y gestión de usuarios y de *cloud storage* y *realtime database* para almacenar el contenido de las unidades didácticas, ejercicios y el resto de información requerida por la app. Más adelante se podría hacer uso de otras funcionalidades como son el envío de notificaciones y las herramientas de análisis, crecimiento y monetización.

1.4.2 Planificación

La asignatura de TFM es de 12 créditos ECTS (*European Credit Transfer System*) y cada crédito ECTS se corresponde a 25 horas de trabajo del estudiante. Por tanto, se dispone de 300 horas para realizar el proyecto.

Las fechas de inicio y fin del proyecto son el 16/09/2020 y el 30/12/2020 respectivamente. Por tanto, el número total de días disponibles entre las dos fechas anteriores es de 106, de los cuales 73 son laborales, 30 son fin de semana y 3 son festivos que no caen en fin de semana. Teniendo en cuenta el recuento anterior la distribución de la dedicación será la siguiente:

Tipo de día	Horas de dedicación	Recuento
Laborable	2,5	73 x 2,5 = 182,5
Fin de semana	4	30 x 4 = 120
Festivo	6	2 x 6 = 12
	TOTAL	314,5

Tabla 14: Horas de dedicación al trabajo

Notas aclaratorias:

1. El festivo correspondiente al 25 de diciembre de 2020 no se dedicará tiempo al trabajo, por lo que no se ha tenido en cuenta.

2. Se asume que se superan en 14,5 horas las 300 horas requeridas.

A continuación se presenta la tabla de hitos del proyecto en la que se puede observar su planificación, costo y estado:

Hito	Inicio	Fin	Dedicación en horas	Estado
PEC1 – Plan de trabajo	16/09	07/10	64	Completado
Contexto y justificación del trabajo	16/09	25/09	28	Completado
Objetivos del trabajo	26/09	27/09	8	Completado
Enfoque y método elegido	28/09	02/10	12,5	Completado
Planificación del trabajo	03/10	07/10	14,5	Completado
Productos obtenidos	07/10	07/10	1	Completado
PEC2 – Diseño	08/10	28/10	65	Completado
Usuarios y contexto de uso	08/10	10/10	8	Completado
Diseño conceptual	10/10	12/10	8	Completado
Prototipado	12/10	18/10	21	Completado
Evaluación: casos de uso	19/10	25/10	20	Completado
Evaluación: diseño de la arquitectura	25/10	28/10	8	Completado
PEC3 – Implementación	29/10	09/12	126,5	Completado
Creación del proyecto base	29/10	30/10	5	Completado
Integración con Firebase	31/10	02/11	10,5	Completado
Registro y login	03/11	06/11	10	Completado
Desarrollo unidades didácticas nivel A1	07/11	12/11	18	Completado
Desarrollo unidades didácticas nivel A2	13/11	18/11	18	Completado
Desarrollo unidades didácticas nivel B1	19/11	24/11	18	Completado
Desarrollo unidades didácticas nivel B2	25/11	30/11	18	Completado
Interacción profesorado-alumnado	01/12	04/12	10	Completado
Pruebas	05/12	09/12	19	Completado
PEC4 – Entrega final	10/12	30/12	59	Completado
Finalizar proyecto	10/12	16/12	20,5	Completado
Memoria	17/12	24/12	23	Completado
Presentación	26/12	27/12	8	Completado
Vídeo	28/12	30/12	7,5	Completado
		TOTAL	314,5	

Tabla 15: Planificación, costo y estado de los hitos del proyecto

En las siguientes figuras se puede observar el diagrama de *Gantt* correspondiente a la planificación temporal de cada una de las prácticas de evaluación continua (PECs) que conforman este trabajo. Comentar que se ha respetado el color utilizado para cada PEC en la tabla de hitos.

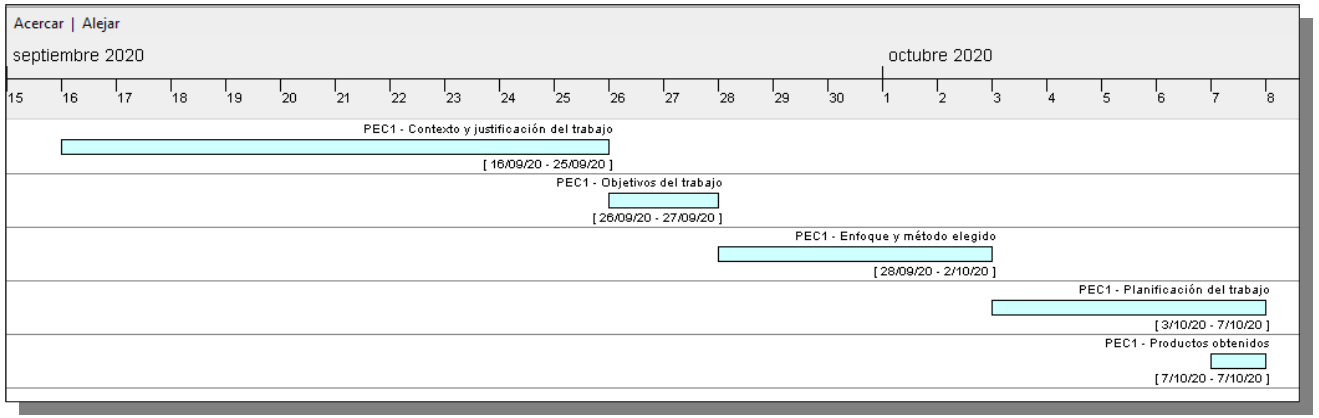

Figura 5: Diagrama de Gantt de la PEC1

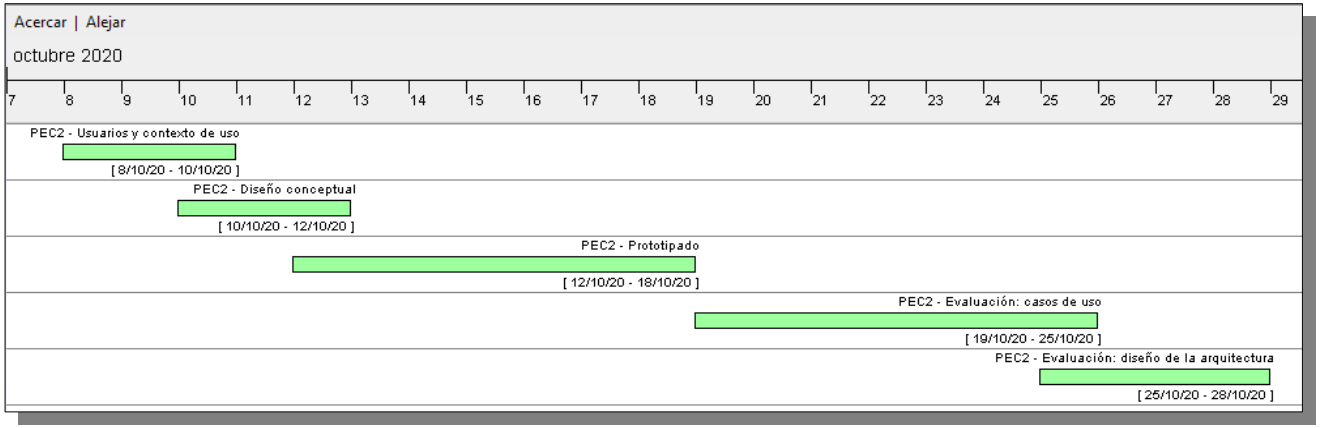

Figura 6: Diagrama de Gantt de la PEC2

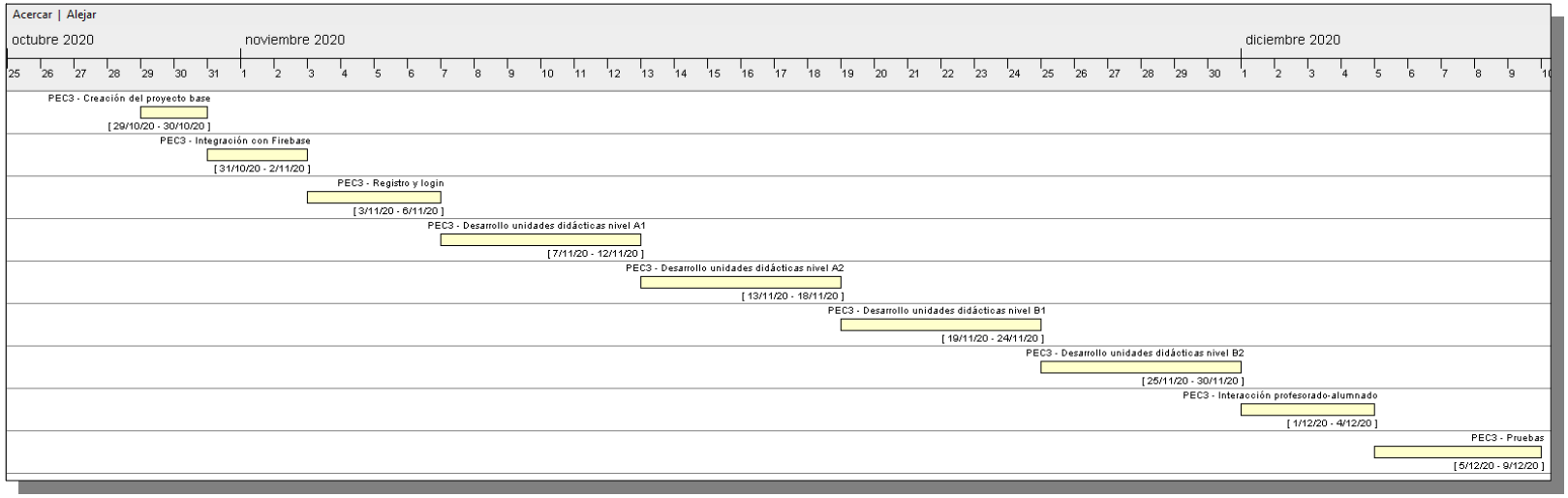

Figura 7: Diagrama de Gantt de la PEC3

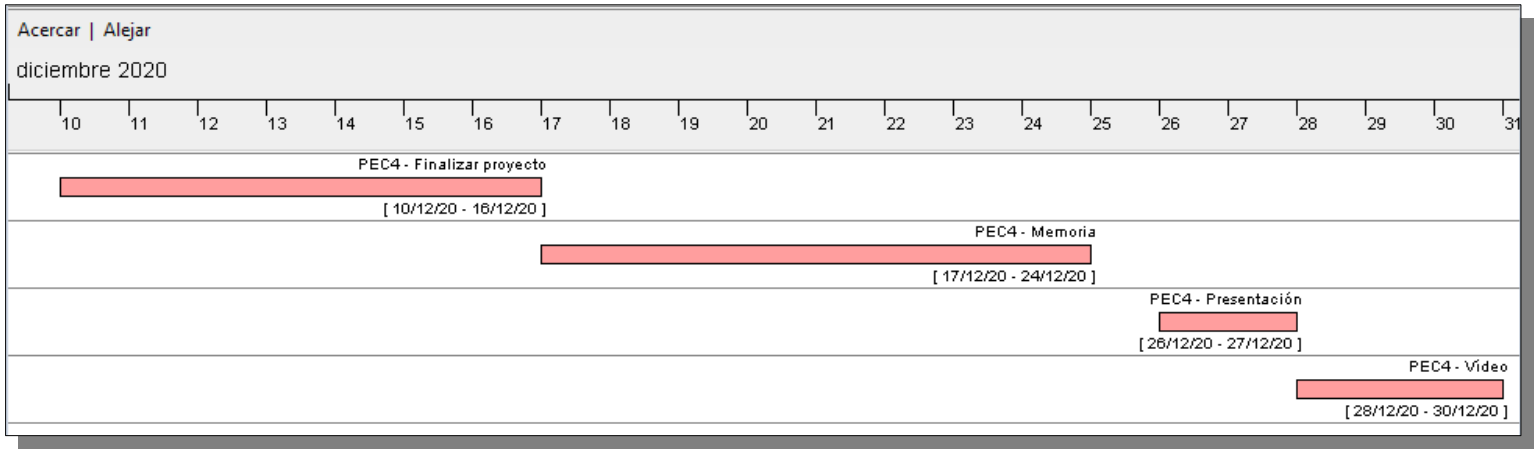

Figura 8: Diagrama de Gantt de la PEC4

1.4.3 Cálculo del coste en el mercado

Tal y como se explica detalladamente en el artículo *¿cuánto cuesta crear una app?* publicado por la plataforma online para la contratación de expertos en tecnología móvil Yeeply (Yeeply, 2020), el coste de una app viene determinado por diversos factores como la tecnología utilizada (nativa, híbrida, web-app), la plataforma destino (Android, iOS), las funcionalidades que ofrece, la complejidad la interfaz de usuario y diseño de la aplicación, etc.

Otro factor a tener en cuenta es si será desarrollada por una empresa o por un desarrollador *freelance*. El precio por hora de un desarrollador *freelance* suele ser más reducido porque no tiene costes estructurales.

El precio por hora también depende de la localidad, ya que según el mencionado artículo de Yeeply, el precio medio por hora, tendiendo en cuenta empresas de desarrollo y profesionales *freelance*, en Sevilla es de 28 €, en Valencia de 39 €, en Barcelona de 42 € y en Madrid de 44 €.

Según el estudio pormenorizado realizado por Yeeply, el coste de desarrollar una aplicación en España tiene una media de 35 € por hora siendo el país donde es más económico desarrollar una app, frente a los 45 € por hora en Italia, los 55 € por hora en Francia y los 84 € por hora en Alemania.

Considerando que se trata de un proyecto que requiere 300 horas de trabajo y que se trata de un desarrollo nativo Android, el precio total de la ejecución del proyecto por parte de:

1. Una empresa de desarrollo con precio medio por hora de 35 € sería de 10.500 €.

2. Un desarrollador freelance con el precio medio por hora de 27,50 € sería de 8.250 €.

1.4.4 Riesgos y medidas de contingencia

En este apartado se describen los principales riesgos y las correspondientes medidas de contingencia que se aplicarían en caso de producirse, con el objetivo de minimizar al máximo su impacto sobre el proyecto.

Pandemia de COVID-19

El impacto sobre el proyecto depende de si se contagia y con qué gravedad quien desarrolla el proyecto, un familiar directo, una buena amistad, o un familiar indirecto. En este caso el único plan de contingencia es intentar no contagiarse evitando riesgos, cumpliendo las normas y siguiendo los consejos de las autoridades competentes. En cualquier caso, es un riesgo que en este momento no es evitable y que lamentablemente se está obligado a asumir, así como sus consecuencias, que en el peor de los casos, sería la cancelación del proyecto.

Problemas de diseño, de implementación o de planificación

Siempre puede surgir un imprevisto que puede afectar en mayor o menor medida a la planificación inicial del proyecto. Si ocurre se intentará recuperar el tiempo de otras tareas pero si no es posible, o bien habrá que invertir horas extraordinarias o bien habrá que renunciar a alguna funcionalidad. En ningún caso se plantea la opción de demorar la entrega del proyecto. En caso de tener que renunciar a alguna funcionalidad, primero se descartarían las menos importantes para las personas usuarias.

Avería, ataque informático o robo de los equipos utilizados para realizar el trabajo

En caso de producirse, el tiempo de recuperación será mayor o menor según el equipo afectado, por ejemplo, si se estropea uno de los dispositivos móviles el impacto no tiene mucha importancia, mientras que si se estropea el ordenador portátil sí. Afortunadamente, actualmente la adquisición de un nuevo equipo es rápida. Además, en este caso, se dispone de un ordenador de sobremesa que podría sustituir al ordenador portátil. En cualquier caso, es básico y fundamental realizar regularmente copias de seguridad, locales y en la nube, de todos los archivos que forman parte del proyecto.

Dependencia de Firebase

Como se ha explicado en el apartado 1.4.1 Entorno de desarrollo, como mínimo se utilizarán las herramientas de autenticación, almacenamiento en la nube y de base de datos en tiempo real de Firebase. Si por algún motivo como, por ejemplo, que Google decida dejar de ofrecer el servicio, no es posible utilizar estas herramientas, se tendrá que optar entre buscar otra solución similar o intentar realizar un desarrollo propio. Afortunadamente, la probabilidad de que se produzca este riesgo es baja.

1.5 Breve sumario de productos obtenidos

Los entregables finales de este trabajo serán:

1. Una aplicación Android nativa correspondiente a una academia virtual de alemán.
    1.1. Código fuente y recursos.
    1.2. Versión empaquetada (APK).
2. La memoria del trabajo en la que se incluye:
    2.1. Manual de usuario de la app desarrollada.
    2.2. Instrucciones de instalación, configuración, compilación y ejecución de la aplicación.
3. Presentación del trabajo en formato PDF.
4. Presentación en vídeo del trabajo.

2. Diseño centrado en el usuario (DCU)

2.1 Usuarios y contexto de uso

En el diseño centrado en el usuario se presentan diversos métodos para obtener información sobre las personas usuarias, sus objetivos y sus necesidades (Flamarich Zampalo, 2017): observación e investigación, seguimiento, diario, entrevistas, dinámicas de grupo, encuestas y análisis de la competencia. De todos ellos, en este trabajo se han utilizado el análisis de la competencia (*benchmarking*) y la entrevista.

2.1.1 Análisis de la competencia

Del análisis de las siete aplicaciones para aprender idiomas realizado en el apartado 1.1.1 Estudio de mercado se han obtenido las siguientes ideas a tener en cuenta en el diseño de la aplicación presentada aquí:

1. Aprendizaje a través de juegos.
2. Combinación de vocabulario con imágenes y sonido.
3. Facilidad de uso a través de ejercicios de selección de la opción correcta, verdadero falso, etc.
4. Posibilidad de utilizar la aplicación sin conexión.
5. Posibilidad de realizar una prueba de nivel (objetivo opcional).
6. Inclusión de un modo demo (objetivo opcional).
7. Sistema de puntos con ranking de usuarios (objetivo opcional).

También se tendrán en cuenta las debilidades de las aplicaciones analizadas con el objetivo de intentar no cometer los mismos errores. Para ello se intentará:

1. Evitar que la aplicación sea monótona.
2. Que la pronunciación en los audios sea correcta.
3. No incordiar excesivamente a la persona usuaria con mensajes o invitaciones.
4. Evitar errores en los contenidos: faltas de ortografía, errores gramaticales, etc.
5. Que la aplicación funcione correctamente, sin bloquearse.

2.1.2 Entrevista

En este apartado se presenta la entrevista semiestructurada que se ha realizado a una profesora nativa alemana. Se trata de una entrevista muy valiosa porque ha aportado información de primera mano sobre la metodología de la enseñanza de alemán, las características y motivaciones de las personas que deciden aprender este idioma y sobre las ventajas e inconvenientes de aprender alemán a través de una app.

A continuación la transcripción de la entrevista:

1. ¿Para qué niveles imparte clases?

   Imparto clases de todos los niveles, desde A1 hasta C2.

2. ¿Dónde imparte sus clases?

   Imparto clases en academias, en escuelas, en empresas y también doy clases particulares.

3. ¿Cuál es el perfil de las personas que deciden estudiar alemán? ¿Cuáles son sus motivaciones?

   Están representados todos los grupos de edad, desde niños de un año hasta personas mayores. Actualmente, mi alumno de mayor edad tiene 77 años. Mucha gente viene para perderle el miedo al idioma, especialmente a la gramática. También hay personas que han empezado a aprender alemán, pero rápidamente se sienten superadas y lo acaban dejando.

   La mayoría de las personas que deciden aprender alemán, lo hacen por motivos profesionales y con el objetivo de tener más posibilidades de encontrar trabajo en España o incluso en los países de habla alemana. Pero también hay casos de personas que deciden aprender alemán por diversión o simplemente para ejercitar su mente.

4. ¿Conoce alguna app para aprender alemán?

   Sí, conozco algunas aplicaciones que ofrecen la posibilidad de aprender alemán, como Babbel, Busuu, etc. También hay muchas aplicaciones para aprender alemán jugando.

   En general, por lo que he visto, la mayoría de estas aplicaciones ofrecen una gran cantidad de lecciones en las que se se tiene la posibilidad de realizar actividades, escuchar diálogos, aprender vocabulario, etc. pero no ofrecen un *feedback* correctivo personalizado y enriquecedor, es decir, el alumno se encuentra solo durante el aprendizaje. En algunas, como en Busuu, se tiene la posibilidad de interactuar con personas nativas, pero en mi opinión no está garantizado un aprendizaje correcto, porque las personas, por ser nativas, no significa que dominen el idioma como para enseñar a otras personas.

5. ¿Cuál es su opinión respecto al aprendizaje de alemán a través de este tipo de apps? ¿Cuáles cree que son sus ventajas e inconvenientes?

   En mi opinión, las aplicaciones son una buena forma de iniciarse en un idioma extranjero como el alemán, ya que permiten aprender vocabulario y hacerse una idea de cómo está estructurado. Además, se pueden utilizar en cualquier sitio y en cualquier momento, pero tienen la desventaja de que el alumno aprende de forma autodidacta y no es consciente de todos los errores que comete, especialmente de los que están relacionados con la expresión oral y escrita.

Toda la información recogida en esta entrevista se ha tenido en cuenta en todas las fases del diseño centrado en el usuario.

2.2 Diseño conceptual

2.2.1 Perfiles de usuario

La versión inicial de la academia virtual de alemán en forma de app presentada aquí, está dirigida a cualquier persona de habla hispana que disponga un dispositivo móvil Android y que tenga interés en aprender alemán a través de un método de aprendizaje personalizado, como lo hace una persona nativa y con *feedback* correctivo. En el futuro se incluirán más colectivos, con idioma base distinto al español como, por ejemplo, las personas de habla inglesa, francesa, etc. El idioma base es el idioma utilizado para la comunicación entre alumnado y profesorado, y en las traducciones incluidas principalmente en los niveles iniciales, que son necesarias cuando las personas usuarias aún no tienen conocimientos de alemán.

Al ofrecerse desde el nivel A1 (principiantes) hasta el nivel B2 (intermedio alto), las personas usuarias pueden ser niños o adolescentes que quieran aprender por iniciativa propia o por la de sus padres con el objetivo de tener mejores perspectivas de futuro, jóvenes y adultos que lo requieran por motivos laborales, o personas mayores a quienes simplemente les apetezca, por ejemplo, como forma de ejercitar su mente.

En cuanto a los contextos de uso de la aplicación, los escenarios de uso son cualquier lugar en el que el usuario disponga de un dispositivo móvil Android con batería y la aplicación instalada. Puntualmente, para ejecutar ciertas funcionalidades, también se requiere de conexión a Internet.

2.2.2 Personas y escenarios

En este apartado se presentan las fichas de personas y escenarios que se han elaborado mediante la herramienta online Xtensio (Xtensio, 2020).

Personas focales:

Antonio López

Fuente: pixabay.com

"Saber hablar varios idiomas te abre un nuevo mundo de posibilidades"

Edad: 30
Trabajo: jefe de recepción
Familia: soltero
Ubicación: Palma (Mallorca)
Carácter: ambicioso

Personalidad

Introvertido	Extrovertido
Insensible	Sensible
Indisciplinado	Disciplinado
Reservado	Abierto
Impaciente	Paciente
Flexible	Obstinado

Nervioso | Amable | Metódico

Objetivos

- Aprender nuevos idiomas.
- Llegar a ser el director de un hotel.
- No estancarse en el trabajo.

Frustraciones

- No entender a una persona a causa del idioma.
- Estudiar pero no aprender.
- Problemas con el sistema informático de reservas.

Biografía

Antonio vive junto a su novia en un piso en Palma, es soltero y trabaja como jefe de recepción en un hotel de cinco estrellas en Alcudia (Mallorca).

Después de finalizar el bachillerato, Antonio estudió un grado de turismo en la universidad de las Islas Baleares. Cuando terminó la carrera y antes de obtener su actual puesto de trabajo, estuvo trabajando durante dos temporadas en distintos departamentos de varios hoteles, con el objetivo de adquirir experiencia y conocimientos prácticos relacionados con sus estudios.

Por otro lado, durante su etapa de estudiante, visitó una academia donde cursó diversos cursos de inglés con el objetivo de practicar y mejorar su pronunciación. Además, vivió durante un año en Londres, donde trabajó como becario en un hotel. Allí su nivel de inglés mejoró exponencialmente.

Actualmente, siempre que puede, intenta aprender y perfeccionar algún idioma más porque es consciente de que en su trabajo es fundamental poderse comunicar correctamente con los clientes del hotel.

Motivaciones

Incentivos
Miedo
Logros
Poder
Social

Marcas

Canales preferidos

Publicidad tradicional
Online y social media
Recomendaciones
Relaciones públicas

Escenario

El pasado 15 de agosto de 2019 Antonio estuvo solo en la recepción. Como de costumbre, llegó un autobús detrás de otro, todos ellos llenos hasta arriba de turistas de muchas nacionalidades diferentes: ingleses, franceses, alemanes, italianos, etc. y Antonio no tuvo problemas para comunicarse con ninguno de ellos para darles la bienvenida, registrarlos, cobrarles, etc. hasta que llegó una pareja de alemanes que se quejaban de que el aire acondicionado de su habitación no funcionaba. Lamentablemente, como Antonio no tenía muchos conocimientos de alemán, no les entendía y, cuando ya se palpaba cierto nerviosismo en el ambiente, la situación se resolvió gracias a la intervención de un joven que pretendía recoger la tarjeta de acceso de su habitación. Tradujo toda la conversación y enseguida Antonio pudo actuar y enviar a un técnico a revisar el aire acondicionado de la habitación de la pareja de alemanes. Como Antonio lo pasó bastante mal, ese mismo día empezó a estudiar alemán a través de una academia virtual en forma de app, en la cual se ofrece aprendizaje personalizado, como lo hace una persona nativa y *feedback* correctivo. Además, al trabajar por turnos, la app se adapta perfectamente a sus necesidades, ya que le permite aprender en cualquier sitio y en cualquier momento.
Hoy, 15 de junio de 2020, Antonio tiene que encargarse de redactar en alemán el protocolo COVID-19 a seguir en el hotel. Afortunadamente, ha aprendido mucho gracias a la academia virtual de alemán en forma de app y ha redactado el protocolo relativamente rápido. Para ello, ha utilizado la app para resolver dudas relativas a tiempos verbales y para enviar el documento para su corrección desde la sección correspondiente.

Lucía Müller

Fuente: pixabay.com

"La mejor forma de aprender un idioma es interactuando con personas nativas"

Edad: 17
Trabajo: estudiante
Familia: soltera
Ubicación: Marratxí (Mallorca)
Carácter: afable

Personalidad

Introvertida	Extrovertida
Insensible	Sensible
Indisciplinada	Disciplinada
Reservada	Abierta
Impaciente	Paciente
Flexible	Obstinada

Divertida | Fiel | Inteligente

Objetivos

- Aprender nuevos idiomas.
- Visitar otros países y conocer su cultura.
- Ser profesora de idiomas.

Frustraciones

- No sacar buenas notas.
- No poder apuntarse a una actividad a causa del horario o del precio.
- No encontrar trabajo.

Biografía

Lucía vive junto a sus padres en una casa de Marratxí, es soltera y está estudiando bachillerato en una escuela privada de Palma.

Desde muy pequeña, le han interesado los idiomas y las diferentes culturas. Además, ha acompañado a sus padres en numerosos viajes a diferentes países. Para ellos es muy importante que Lucía aprenda otros idiomas para tener más posibilidades de encontrar un buen trabajo y, si lo desea, poder vivir en otros países.

Actualmente, Lucía además de español y catalán, tiene conocimientos de alemán porque su padre es alemán y de inglés porque ya con tres años la apuntaron a una escuela inglesa.

Lucía tiene la intención de estudiar el Grado de Magisterio de Educación Primaria con la modalidad de alemán e inglés en Barcelona. También quiere vivir una temporada en Alemania, por lo que tiene pensado participar en un programa de erasmus. Lucía sabe que no va a tener excesivos problemas para afrontar la parte oral pero la parte escrita le preocupa un poco.

Motivaciones

Incentivos
Miedo
Logros
Poder
Social

Marcas

Canales preferidos

Publicidad tradicional
Online y social media
Recomendaciones
Relaciones públicas

Escenario

Es sábado 12 de septiembre de 2020 y Lucía dispone de unas horas para mejorar su expresión escrita en alemán. Para conseguirlo, ya hace tiempo que utiliza la academia virtual de alemán en forma de app, principalmente porque le permite escribir textos de diferente temática y enviarlos para su corrección. Una vez abierta la aplicación, ha seleccionado la sección de expresión escrita del nivel B2 y, en este caso, el ejercicio consiste en escribir un texto de 300 palabras sobre energías renovables. Lucía redacta el texto en la misma app y lo envía para su corrección. Al cabo de dos días recibe la corrección y, por un lado, está contenta porque se le indica que en general el texto está muy bien, pero por otro lado, está algo triste porque sigue cometiendo los mismos errores con los pronombres. Como tiene la tarde libre, decide hacer enseguida los ejercicios de refuerzo que le han propuesto y si le sobra tiempo va a realizar una actividad de compresión lectora y una de comprensión oral, ya que finalmente ha decidido presentarse al examen oficial del nivel B2 de alemán.

Persona secundaria:

Ana Rodríguez

Fuente: pixabay.com

"Nunca dejes de aprender porque la vida nunca te dejará de enseñar."

Edad: 65
Trabajo: profesora de primaria
Familia: viuda con dos hijas
Ubicación: Palma (Mallorca)
Carácter: exploradora

Personalidad

Introvertida	Extrovertida
Insensible	Sensible
Indisciplinada	Disciplinada
Reservada	Abierta
Impaciente	Paciente
Flexible	Obstinada

Abierta | Valiente | Curiosa

Objetivos

- Aprovechar al máximo el tiempo.
- Nunca dejar de aprender.
- Probar nuevas tecnologías.

Frustraciones

- Perder el tiempo.
- Obstáculos que impiden lograr sus objetivos.
- Problemas con la tecnología.

Biografía

Ana vive sola en una casa en Palma de Mallorca, es viuda y trabaja en una escuela privada como profesora de primaria. Tiene dos hijas de 25 y 30 años, que ya han formado sus propias familias, y a las que ayuda siempre que lo necesitan, por ejemplo, cuidando de sus hijos.

Ana es una persona abierta a otros idiomas y culturas, y una de sus mayores pasiones es viajar y pasar un tiempo en otros países. De hecho, durante su etapa de estudiante, vivió una temporada en París para aprender francés tal y como lo hace una persona nativa. Posteriormente, al finalizar sus estudios en Palma, a la edad de 25 años, volvió a Francia, donde trabajó en una escuela católica como profesora de lengua castellana y de francés. Concretamente, vivió en París y fue allí donde conoció a su marido, Pierre. Al cabo de 10 años decidieron mudarse a Palma para formar una familia. Durante todo el tiempo que estuvo junto a Pierre, Ana viajó mucho, por ejemplo, a Londres, donde conoció a su mejor amiga: Harriet y donde perfeccionó sus conocimientos de inglés, lo cuál también resultó ser ventajoso para su trabajo, ya que de vez en cuando también tenía que dar alguna clase en este idioma.

Como Ana este año 2020 se jubila y a causa de la pandemia de COVID-19 no es posible viajar, quiere aprovechar el tiempo y aprender alemán, ya que es un idioma que siempre ha querido aprender para poder conversar con unos amigos que tiene en Berlin y que hace tiempo que quiere visitar junto a su amiga Harriet.

Motivaciones

Incentivos
Miedo
Logros
Poder
Social

Marcas

Canales preferidos

Publicidad tradicional
Online y social media
Recomendaciones
Relaciones públicas

Escenario

Es sábado 10 de octubre de 2020 y hace casi un año que Ana recibió una carta de sus amigos de Berlín que de vez en cuando le escriben para preguntarle qué es de su vida. En aquel momento, como Ana no tenía muchos conocimientos de alemán, se vio obligada a traducirla utilizando el traductor de Google y, aún así, no entendió todo lo que le habían escrito. Como Ana es una persona muy cabezona este hecho no la dejaba tranquila y decidió que quería poder comunicarse en alemán con sus amigos. Además, a causa de la pandemia de COVID-19 estaba pasando bastante tiempo en su casa, por lo que decidió buscar un curso para aprender alemán. Durante su búsqueda encontró una academia virtual de alemán en forma de app, en la que se enseña alemán desde cero y donde durante todo el proceso de aprendizaje se tiene el apoyo de profesorado nativo. Ana decidió instalarla y seguir las unidades didácticas del nivel A1. Actualmente ya se encuentra cursando el nivel A2.
Justamente hoy ha recibido otra carta de sus amigos que, gracias a todo lo que ha aprendido durante los últimos meses, ha podido leer y entender sin problemas. Además, ahora que ya tiene una base ha decidido responderles también en alemán, ya que hasta ahora, ella siempre les había escrito en inglés. Para ello ha abierto la academia virtual en forma de app, ha accedido y repasado la unidad didáctica en la cual se explica cómo escribir una carta, ha redactado la carta y la ha enviado para su corrección desde la sección correspondiente de la app. Ana está muy contenta porque todo son ventajas: sigue aprendiendo alemán y ha escrito su primera carta en este idioma, de la cual, además, recibirá el correspondiente *feedback* correctivo.

2.3 Prototipado

Teniendo en cuenta toda la información recabada en las fases de usuarios y contexto de uso, y de diseño conceptual se ha elaborado:

1. Un sketching (esbozo a mano) rápido utilizando la plantilla de un teléfono móvil (Sketchize, 2020) que ha permitido iterar ideas dinámicamente.
2. Un prototipo de alta fidelidad, que se ha creado utilizando la herramienta JustinMind (JustinMind, 2020).

2.3.1 Sketching

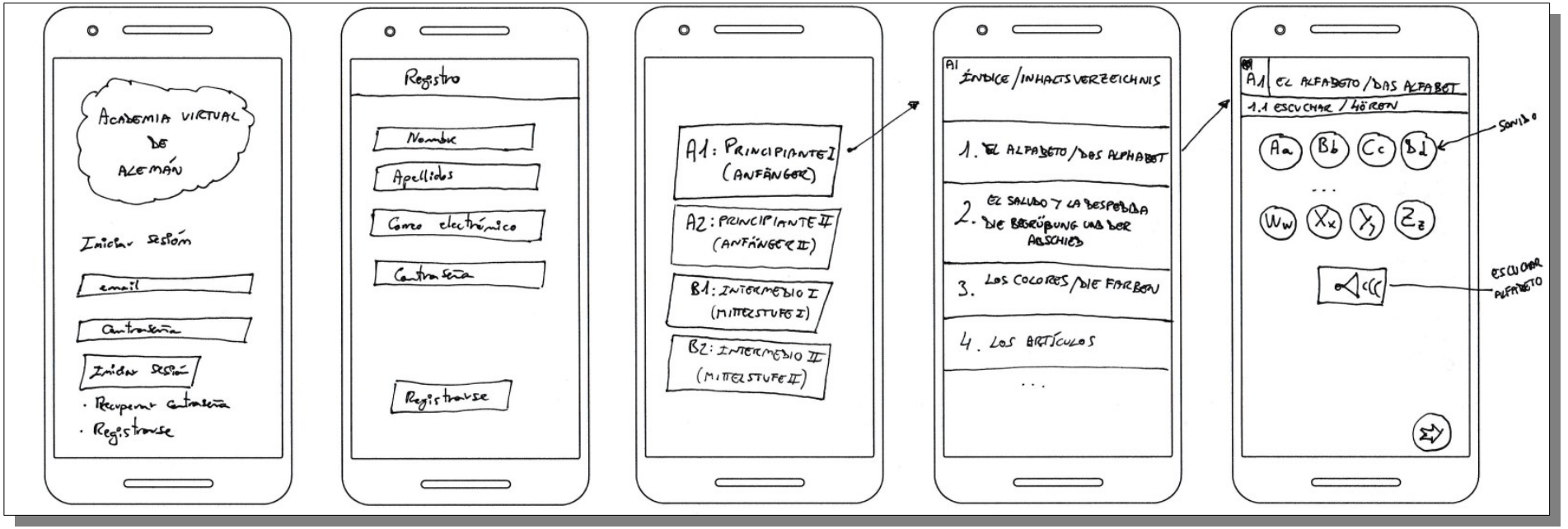

Figura 9: Sketching 1 de 2

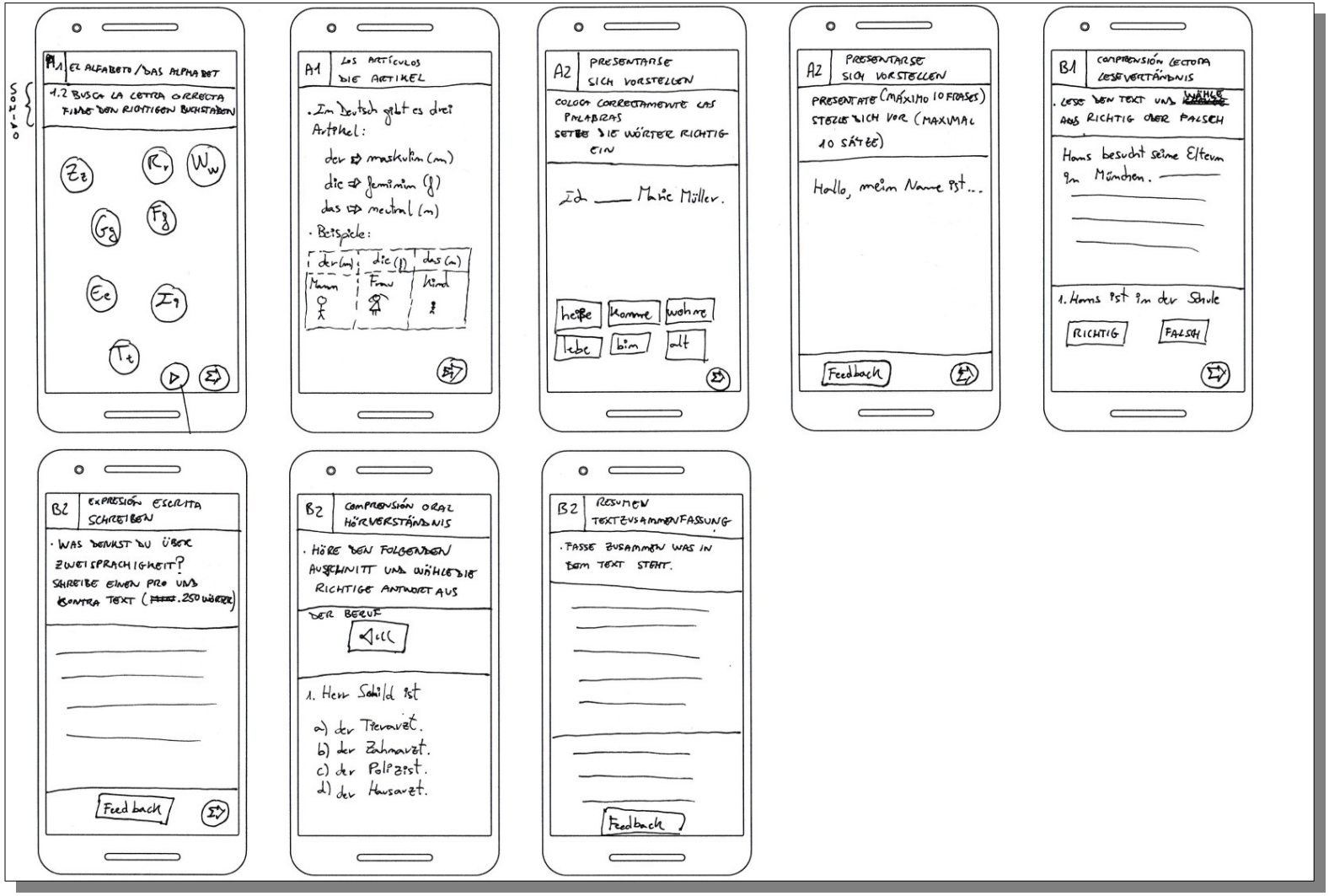

Figura 10: Sketching 2 de 2

2.3.2 Árbol de navegación

En la siguiente figura se presenta el árbol de navegación de la aplicación que se ha creado mediante la herramienta web draw.io (Diagrams.net, 2020) y donde se puede observar que se ha utilizado una combinación de las estructuras de navegación de *hub* y lineal (*nested doll*). Destacar que desde cualquier unidad didáctica se tendrá la posibilidad de navegar a la lista de unidades didácticas del nivel correspondiente.

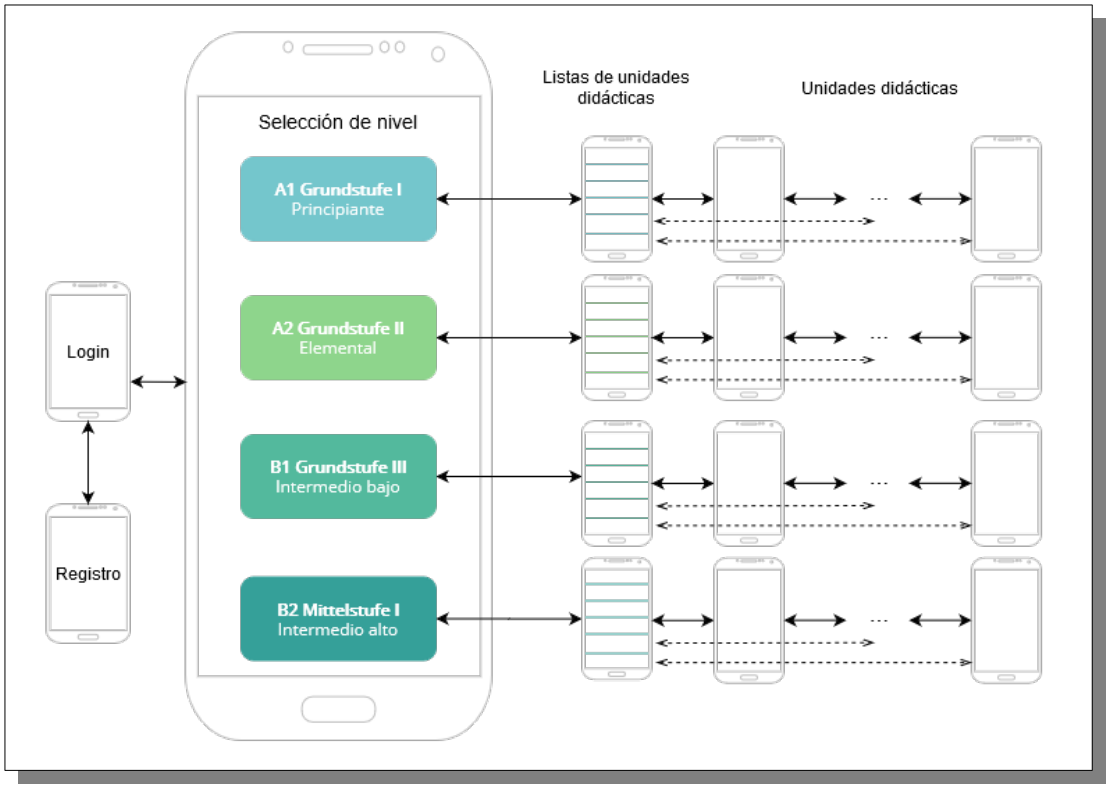

Figura 11: Árbol de navegación

2.3.3 Prototipo de alta fidelidad

A continuación se presenta cada una de las pantallas que componen la versión final del prototipo de alta fidelidad.

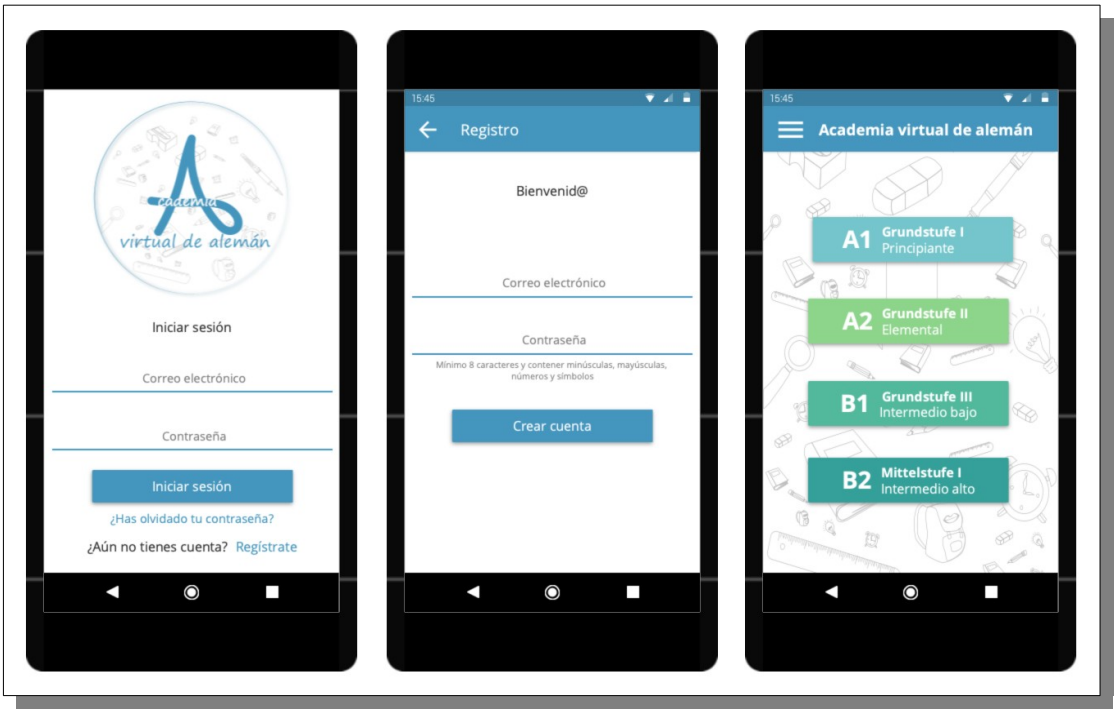

Figura 12: Iniciar sesión - Registro - Selección de nivel de alemán

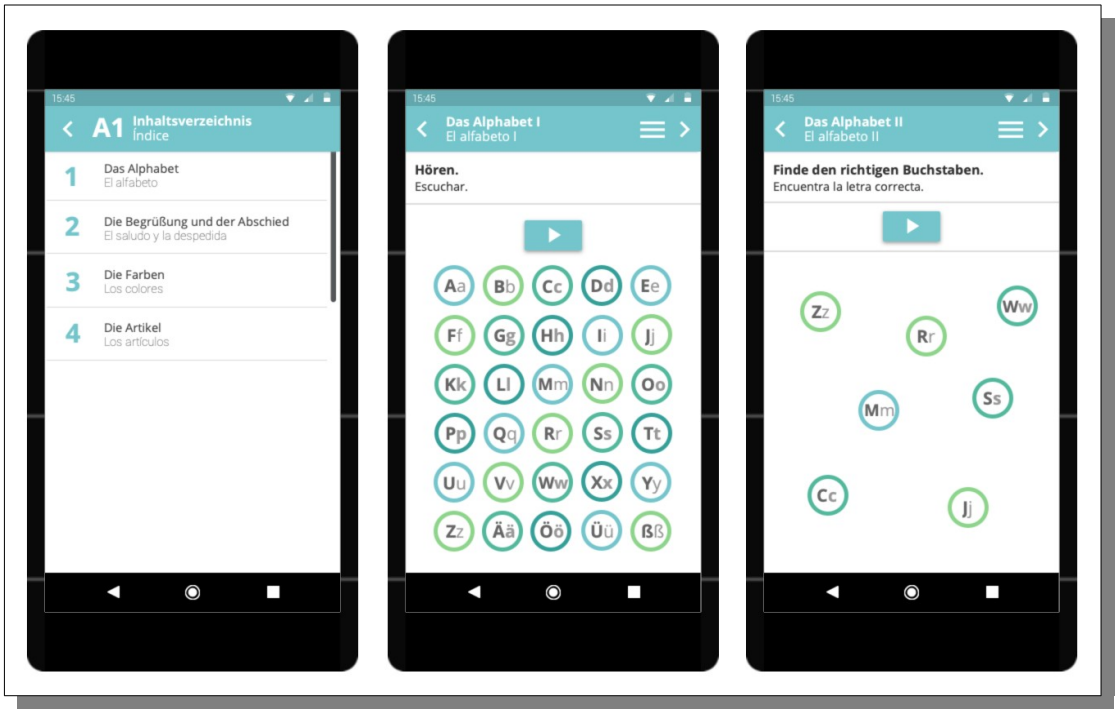

Figura 13: Unidades didácticas del nivel A1 - El alfabeto I - El alfabeto II

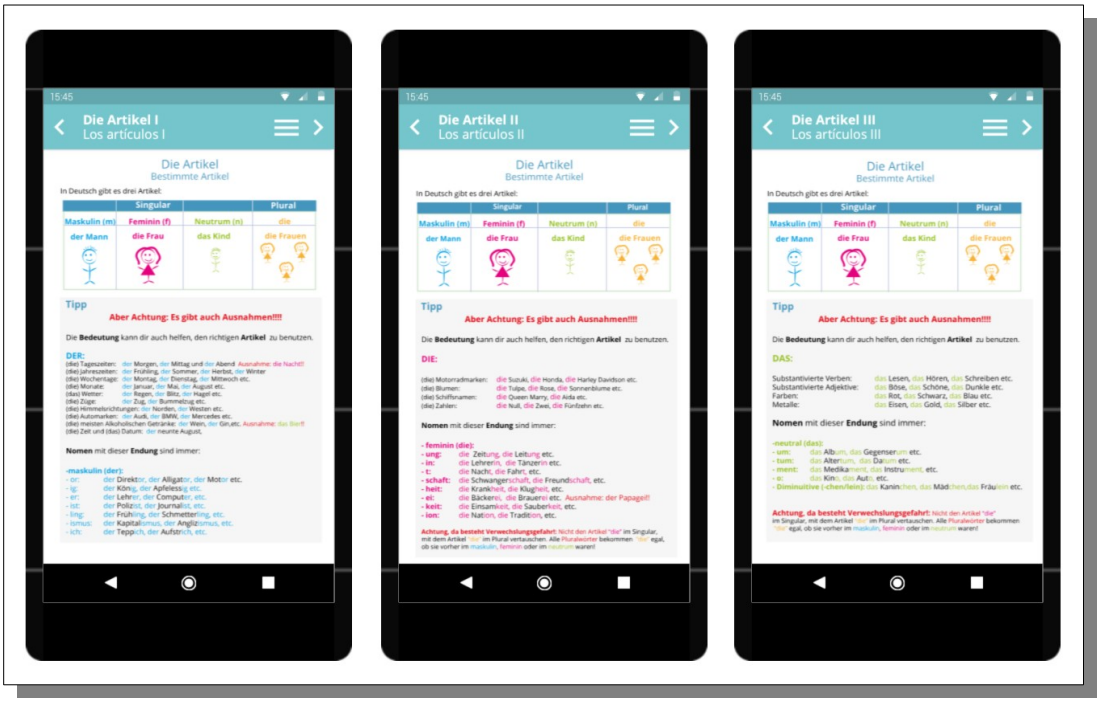

Figura 14: Los artículos I - Los artículos II - Los artículos III

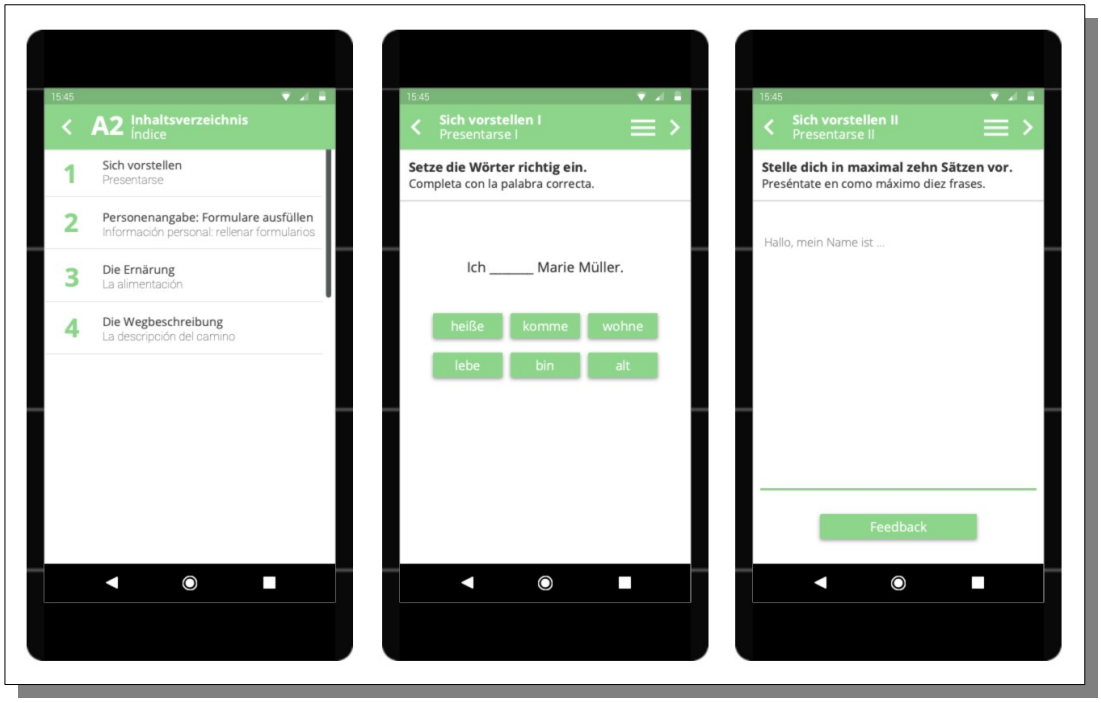

Figura 15: Unidades didácticas del nivel A2 - Presentarse I - Presentarse II

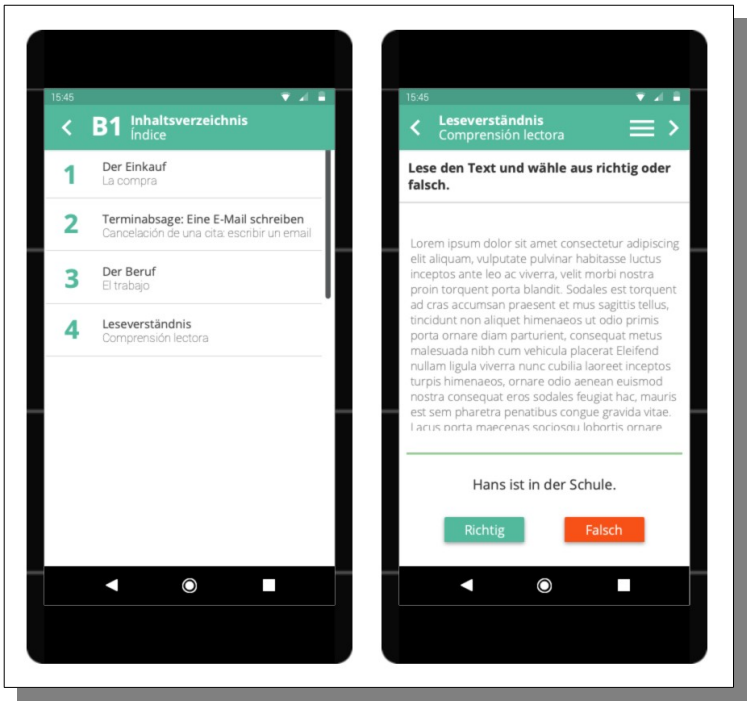

Figura 16: Unidades didácticas del nivel B1 - Comprensión lectora

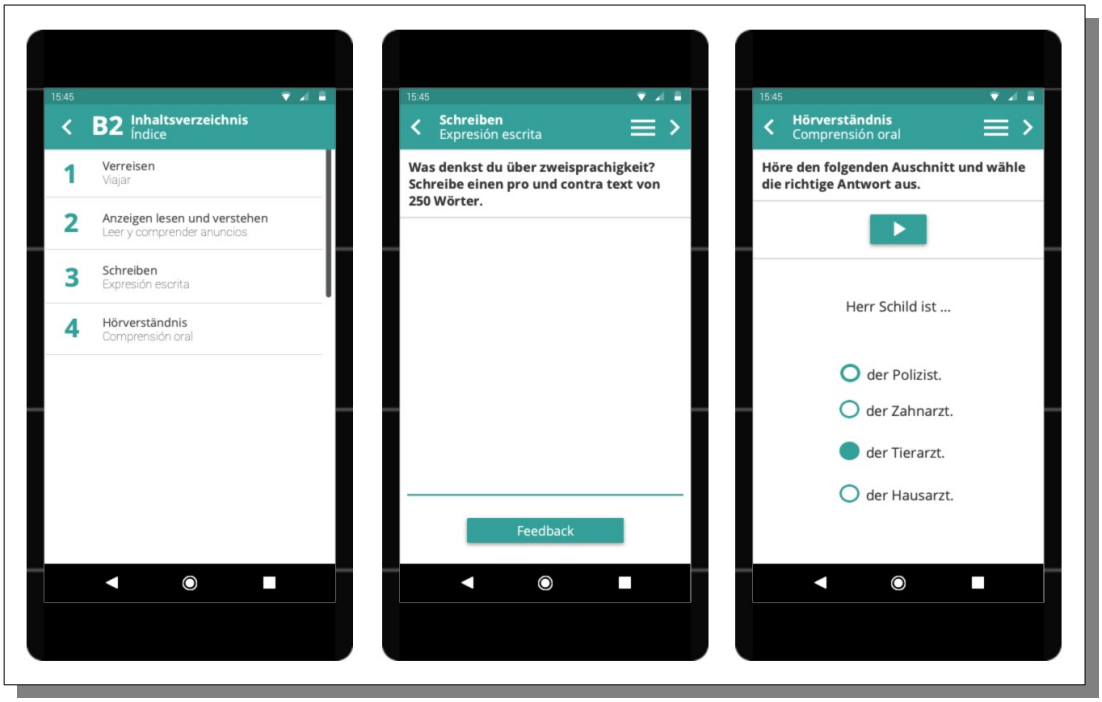

Figura 17: Unidades didácticas del nivel B2 - Expresión escrita - Comprensión oral

2.4 Evaluación

Para evaluar el prototipo de alta fidelidad, se ha presentado a un pequeño grupo de personas formado por una profesora nativa alemana, amigos y familiares. A continuación se presentan las mejoras que se han introducido como resultado de la evaluación.

2.4.1 Evaluación del árbol de navegación

Respecto al árbol de navegación no ha habido nada que objetar excepto que debería ser posible poder acceder al índice o lista de unidades didácticas desde cualquier pantalla correspondiente a una unidad didáctica específica, principalmente, para poder consultar rápidamente otro tema que ya se haya realizado, por ejemplo, para resolver una duda puntual. Sin embargo, no se ha considerado necesario el acceso directo a la pantalla de selección de nivel desde la pantalla de una unidad didáctica específica.

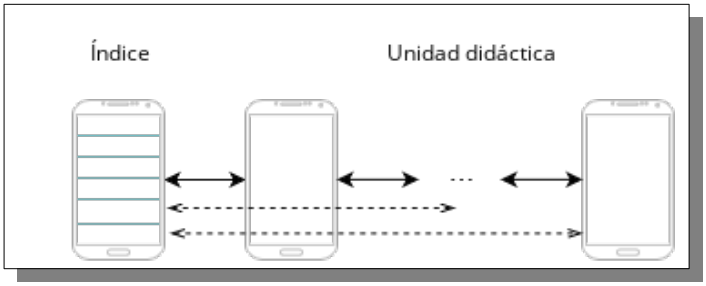

Figura 18: Acceso al índice de unidades didácticas desde cualquier pantalla de una unidad didáctica específica

Para navegar al índice de unidades didácticas desde la pantalla de una unidad didáctica específica, se ha incorporado el botón de menú hamburguesa en el extremo derecho de la barra de tareas.

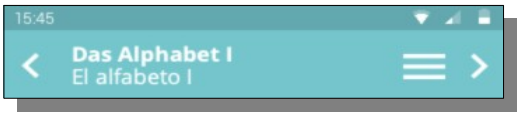

Figura 19: Botón de menú hamburguesa

2.4.2 Evaluación del prototipo de alta fidelidad

Traducciones en el idioma base

La primera cuestión que ha surgido ha sido dónde y en qué medida deben aparecer traducciones en español. Inicialmente, tal y como se puede observar en el 2.3.1 Sketching, se ha priorizado el español respecto al alemán en botones, títulos, enunciados, etc. Tras analizarlo con las personas evaluadoras, se ha considerado que al tratarse de una academia virtual de alemán, es

preferible priorizar el alemán respecto al español y que el uso de este último debería reducirse a medida que se avanza de nivel. La principal argumentación ha sido que si bien es cierto que las traducciones al español son necesarias para las personas principiantes, en las academias físicas se utilizan lo mínimo posible. Finalmente, comentar que además de priorizar el idioma alemán, cuando ha sido posible, se ha enfatizado escribiéndolo en negrita.

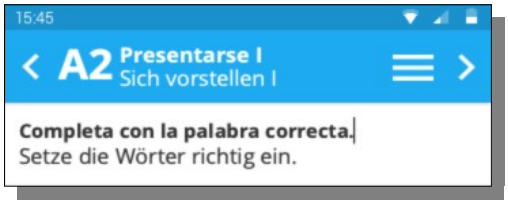

Figura 20: Prototipo inicial: español como idioma preferente

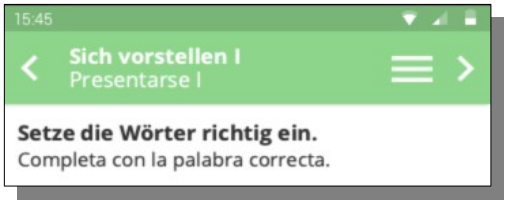

Figura 21: Prototipo final: alemán como idioma preferente

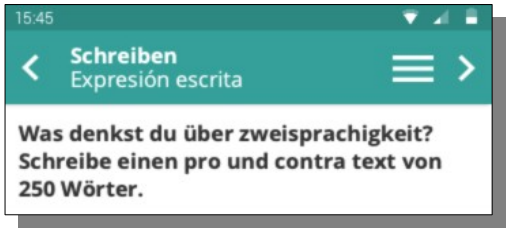

Figura 22: En el nivel B2 ya no consta traducción en español del enunciado

Visualización del nivel de idioma actual

En las primeras iteraciones del prototipo toda la interfaz de usuario se había diseñado en color azul y blanco. Se había elegido el color azul por ser el color favorito de la mayoría de las personas del mundo (YouGov, 2015). Por otro lado, para que el usuario fuera consciente del nivel de idioma actual, éste se visualizaba en la barra de tareas. A las personas evaluadoras, de entrada no les ha parecido mal, pero cuando ha surgido la propuesta de utilizar un color distinto para cada nivel, la han preferido y han manifestado que en este caso no es necesario indicar explícitamente el nivel en la barra de tareas.

Esta propuesta ha provocado la elección de un color para cada nivel, tarea que no ha resultado sencilla, ya que seleccionar cuatro colores más un color principal que combinen no es trivial. Finalmente, se ha utilizado una combinación de azules y verdes que, tienen un efecto relajante y tranquilizador.

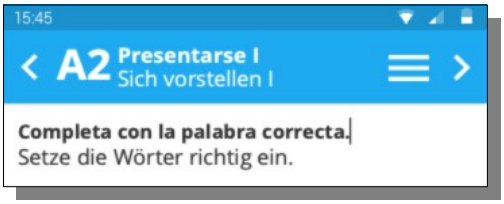

Figura 23: Prototipo inicial, colores azul y blanco para todos los niveles e indicación de nivel en todas las pantallas

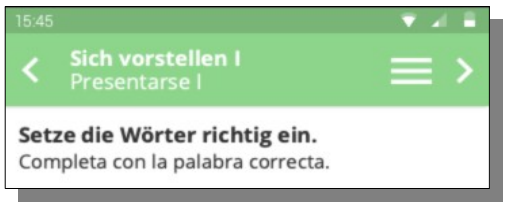

Figura 24: Prototipo final, uso de un color distinto para cada nivel y sin indicación de nivel en todas las pantallas

Figura 25: Uso de un color distinto para cada nivel

Ubicación de los botones en la barra de tareas

En la primera versión del prototipo, en las pantallas correspondientes a una unidad didáctica, se había ubicado la opción para regresar a la lista o índice de unidades didácticas del nivel actual en el extremo izquierdo de la barra de tareas. Además, en cada extremo de la parte inferior de la pantalla se habían ubicado los botones para avanzar y retroceder dentro de la unidad didáctica actual. Las personas evaluadoras, tras probarlo, han indicado que sería más adecuado ubicar los botones para avanzar y retroceder en la unidad didáctica en cada extremo de la barra de tareas y colocar el botón para volver al índice de unidades didácticas en el extremo derecho de la barra de tareas, a la izquierda del botón para avanzar dentro de la unidad didáctica actual. De este modo, todas las navegaciones quedan centralizadas en la barra de tareas y no ocupan espacio de la zona reservada para el contenido de la unidad didáctica.

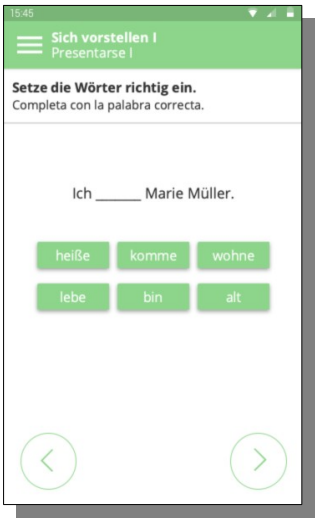

Figura 26: Prototipo inicial, botones en la parte inferior de la pantalla

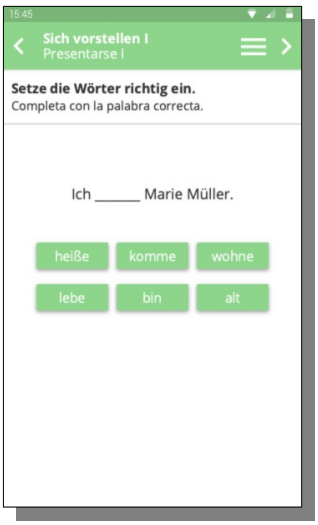

Figura 27: Prototipo final, botones en la barra de tareas

3. Diseño técnico de la aplicación

3.1 Definición de los casos de uso

Los casos de uso son utilizados para describir, desde el punto de vista del usuario, el comportamiento del sistema en forma de acciones y reacciones, definiendo sus límites y sus reacciones con el entorno.

En la siguiente figura, creada mediante la herramienta draw.io (Diagrams.net, 2020) se presenta el diagrama de los casos de uso identificados:

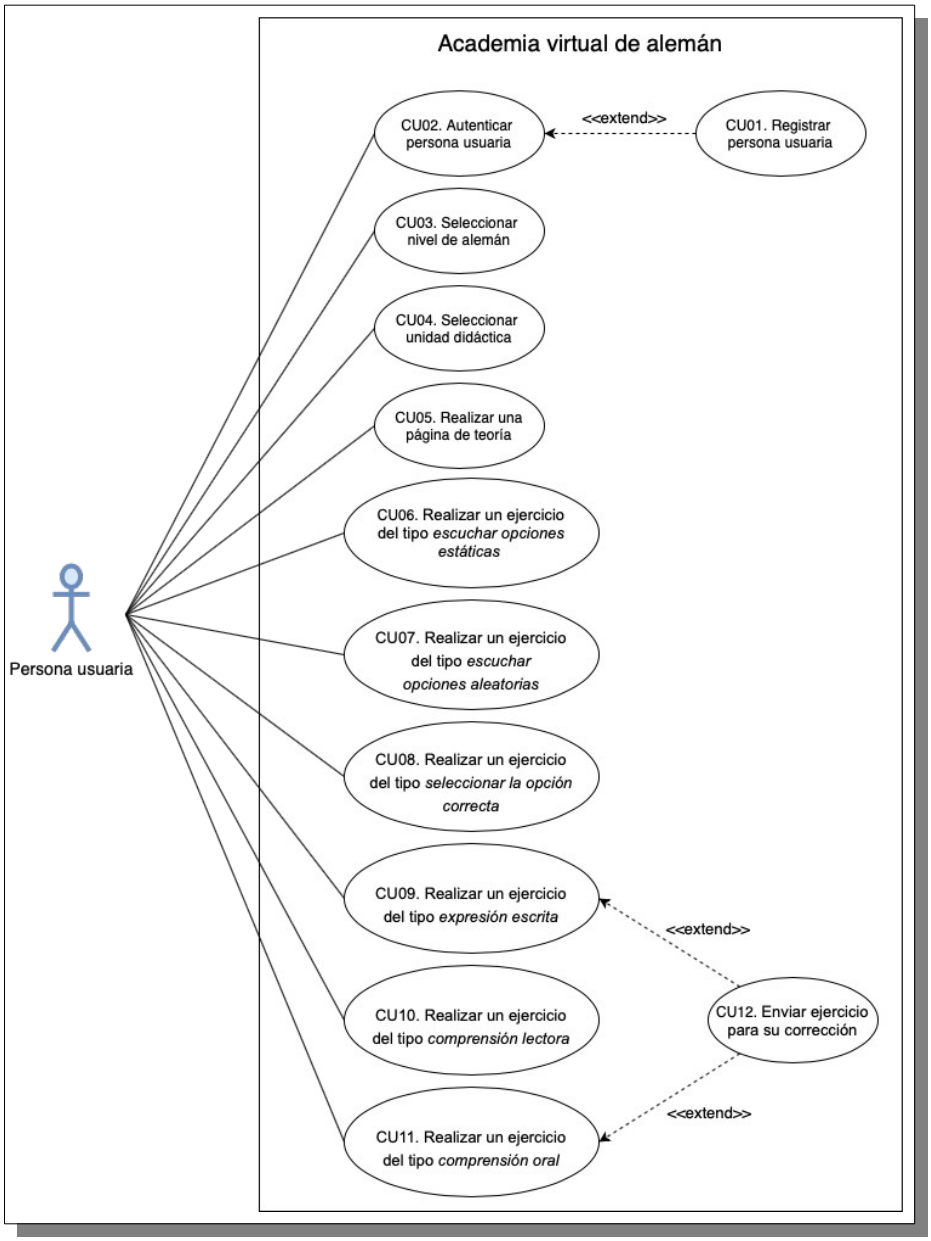

Figura 28: Diagrama de los casos de uso

A continuación la representación textual de los casos de uso identificados:

CU01. Registrar persona usuaria		
Descripción	El sistema ha de permitir que una persona usuaria se registre.	
Precondición	El sistema se encuentra en un estado consistente en la pantalla *iniciar sesión*.	
Postcondición	El sistema registra a la persona usuaria y la redirige a la pantalla *seleccionar nivel de alemán*.	
Postcondición en caso de fracaso	El sistema informa del error y pasa a un estado consistente.	
Actores	Persona usuaria.	
Trigger	La persona usuaria activa la opción *Regístrate* de la pantalla *iniciar sesión*.	
Secuencia normal	**Paso**	**Acción**
	1	Especificar una dirección de correo electrónico.
	2	Especificar una contraseña.
	3	Pulsar el botón *Crear cuenta*.
Observaciones	La dirección de correo electrónico no debe existir en el sistema. La contraseña ha de tener como mínimo 6 caracteres y contener letras y números.	

Tabla 16: Caso de uso CU01. Registrar persona usuaria

CU02. Autenticar persona usuaria		
Descripción	El sistema ha de permitir que una persona usuaria se autentique.	
Precondición	El sistema se encuentra en un estado consistente en la pantalla *iniciar sesión*.	
Postcondición	El sistema ha autenticado a la persona usuaria y la redirige a la pantalla de *seleccionar nivel de alemán*.	
Postcondición en caso de fracaso	El sistema informa del error y pasa a un estado consistente.	
Actores	Persona usuaria.	
Trigger	La persona usuaria ha iniciado la aplicación o acaba de cerrar sesión.	
Secuencia normal	**Paso**	**Acción**
	1	Especificar la dirección de correo electrónico.
	2	Especificar la contraseña.
	3	Pulsar el botón *Iniciar sesión*.
Observaciones		

Tabla 17: Caso de uso CU02. Autenticar persona usuaria

CU03. Seleccionar nivel de alemán		
Descripción	El sistema ha de permitir seleccionar un nivel de alemán.	
Precondición	La persona usuaria se ha autenticado correctamente. El sistema se encuentra en un estado consistente en la pantalla *seleccionar nivel de alemán*.	
Postcondición	La persona usuaria ha seleccionado un nivel de alemán y el sistema la redirige a la pantalla de *índice de unidades didácticas* del nivel seleccionado.	
Postcondición en caso de fracaso	El sistema informa del error y pasa a un estado consistente.	
Actores	Persona usuaria.	
Trigger	La persona usuaria se ha autenticado correctamente o ha regresado de una pantalla de *índice de unidades didácticas* de un determinado nivel.	
Secuencia normal	**Paso**	**Acción**
	1	Seleccionar el nivel de alemán deseado.
Observaciones	Los niveles de alemán disponibles son: A1, A2, B1 y B2.	

Tabla 18: Caso de uso CU03. Seleccionar nivel de alemán

CU04. Seleccionar unidad didáctica		
Descripción	El sistema ha de permitir seleccionar una unidad didáctica.	
Precondición	El sistema se encuentra en un estado consistente en la pantalla de *índice de unidades didácticas* del nivel que haya seleccionado la persona usuaria previamente.	
Postcondición	La persona usuaria ha seleccionado una unidad didáctica y el sistema la redirige a la primera pantalla de la misma.	
Postcondición en caso de fracaso	El sistema informa del error y pasa a un estado consistente.	
Actores	Persona usuaria.	
Trigger	La persona usuaria ha seleccionado un nivel de alemán o ha regresado de una unidad didáctica seleccionada previamente.	
Secuencia normal	**Paso**	**Acción**
	1	Seleccionar la unidad didáctica deseada.
Observaciones		

Tabla 19: Caso de uso CU04. Seleccionar unidad didáctica

CU05. Realizar una página de teoría		
Descripción	El sistema ha de permitir realizar una página de teoría.	
Precondición	El sistema se encuentra en un estado consistente en la pantalla correspondiente a una página de teoría.	
Postcondición	La persona usuaria ha leído y entendido la página de teoría.	
Postcondición en caso de fracaso	El sistema informa del error y pasa a un estado consistente.	
Actores	Persona usuaria.	
Trigger	La persona usuaria ha seleccionado una unidad didáctica en la que consta una página de teoría, a la cual ha accedido.	
Secuencia normal	**Paso**	**Acción**
	1	Leer y entender el contenido de la página de teoría.
Observaciones		

Tabla 20: Caso de uso CU05. Realizar una página de teoría

CU06. Realizar un ejercicio del tipo *escuchar opciones estáticas*		
Descripción	El sistema ha de permitir realizar un ejercicio del tipo *escuchar opciones estáticas*.	
Precondición	El sistema se encuentra en un estado consistente en la pantalla correspondiente a un ejercicio del tipo *escuchar opciones estáticas*.	
Postcondición	La persona usuaria ha escuchado todas las opciones estáticas y/o las que ha seleccionado pulsando sobre ellas.	
Postcondición en caso de fracaso	El sistema informa del error y pasa a un estado consistente.	
Actores	Persona usuaria.	
Trigger	La persona usuaria ha seleccionado una unidad didáctica en la que consta un ejercicio del tipo *escuchar opciones estáticas*, al cual ha accedido.	
Secuencia normal	**Paso**	**Acción**
	1	Seleccionar la opción que permite escuchar todas las opciones estáticas.
	2	Seleccionar la opción estática que se desea escuchar.
Observaciones		

Tabla 21: Caso de uso CU06. Realizar un ejercicio del tipo escuchar opciones estáticas

CU07. Realizar un ejercicio del tipo *escuchar opciones aleatorias*		
Descripción	El sistema ha de permitir realizar un ejercicio del tipo *escuchar opciones aleatorias*.	
Precondición	El sistema se encuentra en un estado consistente en la pantalla correspondiente a un ejercicio del tipo *escuchar opciones aleatorias*.	
Postcondición	La persona usuaria ha escuchado todas las opciones aleatorias. ha pulsado sobre las que le ha parecido entender y el sistema le ha indicado si su elección ha sido correcta o incorrecta.	
Postcondición en caso de fracaso	El sistema informa del error y pasa a un estado consistente.	
Actores	Persona usuaria.	
Trigger	La persona usuaria ha seleccionado una unidad didáctica en la que consta un ejercicio del tipo *escuchar opciones aleatorias*, al cual ha accedido.	
Secuencia normal	**Paso**	**Acción**
	1	Seleccionar la opción que permite escuchar todas las opciones aleatorias.
	2	En la pantalla van apareciendo opciones de forma aleatoria y cada una de ellas perdura unos segundos. Paralelamente, también de forma aleatoria, se escucha una de las opciones.
	3	Seleccionar las opciones que se cree haber entendido.
Observaciones		

Tabla 22: Caso de uso CU07. Realizar un ejercicio del tipo escuchar opciones aleatorias

CU08. Realizar un ejercicio del tipo *seleccionar la opción correcta*		
Descripción	El sistema ha de permitir realizar un ejercicio del tipo *seleccionar la opción correcta.*	
Precondición	El sistema se encuentra en un estado consistente en la pantalla correspondiente a un ejercicio del tipo *seleccionar la opción correcta.*	
Postcondición	La persona usuaria ha seleccionado la opción que ha considerado correcta en cada una de las cuestiones presentadas de forma secuencial por el sistema y ha sido informado de si su elección ha sido correcta o incorrecta.	
Postcondición en caso de fracaso	El sistema informa del error y pasa a un estado consistente.	
Actores	Persona usuaria.	
Trigger	La persona usuaria ha seleccionado una unidad didáctica en la que consta un ejercicio del tipo *seleccionar la opción correcta*, al cual ha accedido.	
Secuencia normal	**Paso**	**Acción**
	1	Leer la cuestión y seleccionar la opción que se considera correcta.
	2	El sistema presenta la siguiente cuestión.
	3	Repetir los pasos 2 y 3 hasta completar todas las cuestiones presentadas por el sistema.
Observaciones		

Tabla 23: Caso de uso CU08. Realizar un ejercicio del tipo seleccionar la opción correcta

CU09. Realizar un ejercicio del tipo *expresión escrita*		
Descripción	El sistema ha de permitir realizar un ejercicio del tipo *expresión escrita.*	
Precondición	El sistema se encuentra en un estado consistente en la pantalla correspondiente a un ejercicio del tipo expresión escrita.	
Postcondición	La persona usuaria ha redactado el texto requerido en el enunciado del ejercicio y, opcionalmente, lo ha enviado para su corrección.	
Postcondición en caso de fracaso	El sistema informa del error y pasa a un estado consistente.	
Actores	Persona usuaria.	
Trigger	La persona usuaria ha seleccionado una unidad didáctica en la que consta un ejercicio del tipo *expresión escrita*, al cual ha accedido.	
Secuencia normal	**Paso**	**Acción**
	1	Redactar el texto requerido en el enunciado.
	2	Opcionalmente, pulsar el botón *feedback* para enviar el texto para su corrección y obtención del correspondiente *feedback* correctivo (CU12).
Observaciones		

Tabla 24: Caso de uso CU09. Realizar un ejercicio del tipo expresión escrita

CU10. Realizar un ejercicio del tipo *comprensión lectora*		
Descripción	El sistema ha de permitir realizar un ejercicio del tipo *comprensión lectora.*	
Precondición	El sistema se encuentra en un estado consistente en la pantalla correspondiente a un ejercicio del tipo comprensión lectora.	
Postcondición	La persona usuaria ha leído el texto del enunciado del ejercicio, ha seleccionando la opción que ha considerado correcta en cada una de las cuestiones presentadas de forma secuencial por el sistema y ha sido informado de si su elección ha sido correcta o incorrecta.	
Postcondición en caso de fracaso	El sistema informa del error y pasa a un estado consistente.	
Actores	Persona usuaria.	
Trigger	La persona usuaria ha seleccionado una unidad didáctica en la que consta un ejercicio del tipo compren*sión lectora*, al cual ha accedido.	
Secuencia normal	**Paso**	**Acción**
	1	Lectura del texto del enunciado del ejercicio.
	2	Leer la cuestión y seleccionar la opción que se considera correcta.
	3	El sistema presenta la siguiente cuestión.
	4	Repetir los pasos 2 y 3 hasta completar todas las cuestiones presentadas por el sistema.
Observaciones		

Tabla 25: Caso de uso CU10. Realizar un ejercicio del tipo comprensión lectora

CU11. Realizar un ejercicio del tipo *comprensión oral*		
Descripción	El sistema ha de permitir realizar un ejercicio del tipo *comprensión oral.*	
Precondición	El sistema se encuentra en un estado consistente en la pantalla correspondiente a un ejercicio del tipo comprensión oral.	
Postcondición	La persona usuaria ha escuchado el audio del ejercicio, ha seleccionado la opción que ha considerado correcta para cada una de las cuestiones presentadas de forma secuencial por el sistema y ha sido informado de si su elección ha sido correcta o incorrecta. A continuación ha respondido por escrito a otro conjunto de preguntas que, opcionalmente, ha enviado para su corrección.	
Postcondición en caso de fracaso	El sistema informa del error y pasa a un estado consistente.	
Actores	Persona usuaria.	
Trigger	La persona usuaria ha seleccionado una unidad didáctica en la que consta un ejercicio del tipo compren*sión oral*, al cual ha accedido.	
Secuencia normal	**Paso**	**Acción**
	1	Escuchar el audio una o dos veces.
	2	Leer la cuestión y seleccionar la opción que se considera correcta.
	3	El sistema presenta la siguiente cuestión.
	4	Repetir los pasos 2 y 3 hasta completar todas las cuestiones presentadas por el sistema.
	5	Contestar por escrito a las preguntas relacionadas con el texto presentadas por el sistema y, al acabar y opcionalmente, pulsar el botón *feedback* para enviar las respuestas para su corrección y obtención del correspondiente *feedback* correctivo (CU12).
Observaciones		

Tabla 26: Caso de uso CU11. Realizar un ejercicio del tipo comprensión oral

CU12. Enviar ejercicio para su corrección		
Descripción	El sistema ha de permitir enviar un ejercicio para su corrección.	
Precondición	El sistema se encuentra en un estado consistente en la pantalla de un ejercicio que puede ser enviado para ser corregido. El ejercicio está completado.	
Postcondición	El ejercicio ha sido enviado para ser corregido.	
Postcondición en caso de fracaso	El sistema informa del error y pasa a un estado consistente.	
Actores	Persona usuaria.	
Trigger	La persona usuaria ha seleccionado una unidad didáctica en la que consta un ejercicio que puede ser enviado para ser corregido, al cual ha accedido y ha completado.	
Secuencia normal	**Paso**	**Acción**
	1	Pulsar sobre la opción para enviar a corregir el ejercicio.
Observaciones		

Tabla 27: Caso de uso CU12. Enviar ejercicio para su corrección

3.2 Diseño de la arquitectura

En la siguiente figura se presenta la arquitectura del sistema. Como se explicó en el apartado 1.4.1 Entorno de desarrollo se utilizará Firebase como backend, concretamente la funcionalidad de *authentication* para la autenticación y gestión de usuarios y de *cloud storage* y *realtime database* para almacenar y distribuir el contenido de las unidades didácticas, ejercicios y el resto de información requerida por la aplicación.

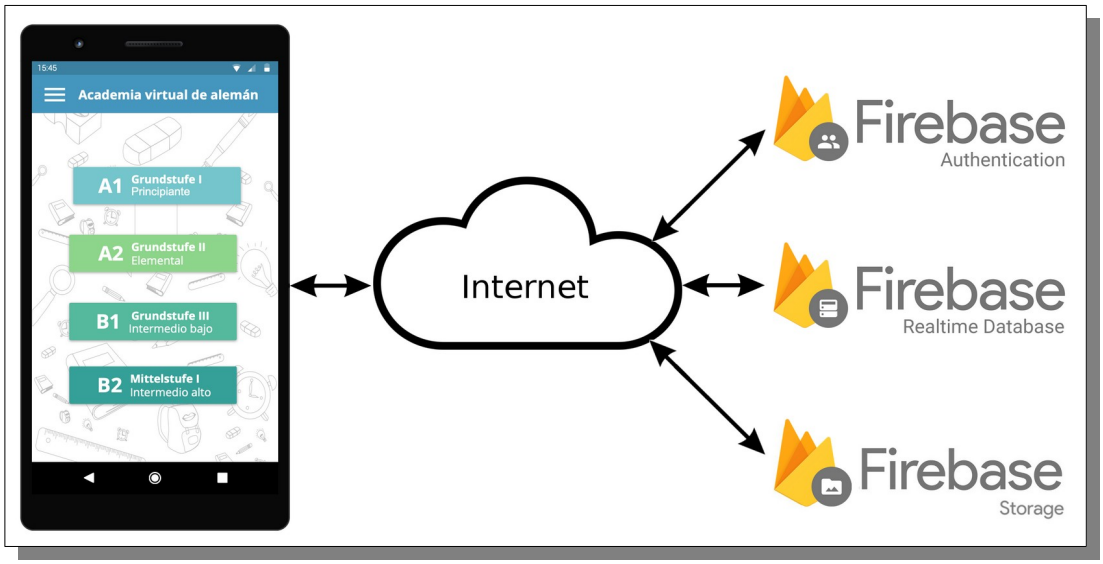

Figura 29: Arquitectura del sistema

3.2.1 Diagrama de clases

La funcionalidad *realtime database* de la plataforma en la nube Firebase que se utilizará para almacenar las unidades didácticas, es una base de datos NoSQL donde, como en la mayoría de sistemas NoSQL, los datos se organizan como agregados. Concretamente, en este caso, los datos utilizan el formato JSON (*JavaScript Object Notation*).

En la figura 30 se presenta el diagrama de clases UML que, como se puede observar, excepto para la clase Credits, se corresponde a una jerarquía de agregación, donde una entidad agrega recursivamente a otras entidades.

Descripción de las clases:

Level: primera clase de la jerarquía de agregación correspondiente a los niveles de alemán.

DidacticUnit: contiene las unidades didácticas de un determinado nivel.

Page: cada una de las páginas que conforman una unidad didáctica como, por ejemplo, un ejercicio de comprensión lectora. El atributo type permite especificar el tipo de página: de teoría o de ejercicios.

Content: contenidos de una página como, por ejemplo, un ejercicio de seleccionar la opción correcta. El atributo type permite especificar el tipo de contenido: texto, audio, etc.

Options: posibles opciones de un determinado contenido como, por ejemplo, las de un ejercicio de seleccionar la opción correcta.

Credits: clase independiente que contiene los créditos de la aplicación.

Las clases Content y Options contienen los atributos imagePath y audioPath que indican la ruta a un posible recurso de imagen o audio almacenados en la nube (Firebase Cloud Storage).

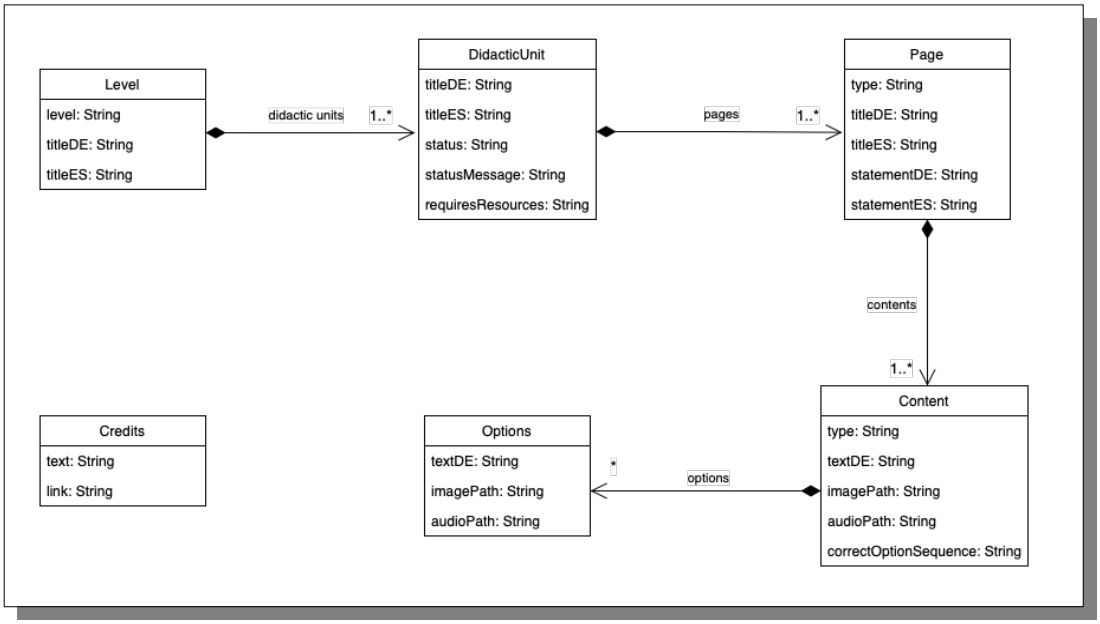

Figura 30: Diagrama de clases UML

Finalmente, avanzar que en el apartado 4.2 de este documento, se presentará la transformación del diagrama de clases en una estructura de nodos JSON con el objetivo de ejemplificar las características de los datos insertados en la base de datos NoSQL.

3.2.2 Patrón de arquitectura de software

MVP (Model View Presenter), MVVM (Model-View ViewModel), VIPER (View, Interactor, Presenter, Entity and Router) y MVI (Model View Intent) son las cuatro arquitecturas más utilizadas en Android para conseguir mantener el código organizado, gestionado y testado. Todas ellas son evoluciones de MVC (Model View Controller), que fue una de las primeras soluciones consistentes en separar la lógica de negocio de la interfaz de usuario.

Para el desarrollo de la aplicación presentada aquí se utilizará el patrón de diseño MVP, que es uno de los patrones de arquitectura más comunes y usados en el desarrollo nativo de Android. Se ha elegido la arquitectura MVP porque se ha considerado que es la que mejor se ajusta a la complejidad del proyecto y de las pantallas a realizar. Las arquitecturas MVVM, VIPER y MVI son más adecuadas para proyectos más complejos y de mayor envergadura que el de este trabajo.

En la arquitectura MVP se logra separar la vista del controlador, lo cuál no es posible en la arquitectura Model View Controller (MVC) donde la *Activity* sirve tanto la vista como el controlador.

La arquitectura MVP está compuesta por:

Model: es la capa de datos responsable de la lógica de negocio.

View: visualiza la interfaz de usuario y posibilita la interacción con el usuario.

Presenter: se comunica con el modelo y con las vistas permitiendo conectar la interfaz gráfica con los datos.

Respecto al flujo, en la siguiente figura se puede observar que a diferencia de MVC, donde el punto de entrada de la aplicación es el controlador, en MVP el punto de entrada es la vista. La vista comunica con el presentador y éste actualiza el modelo. Una vez actualizado el modelo el presentador actualiza la vista. De este modo, se consigue que vista y modelo sean totalmente independientes entre ellos.

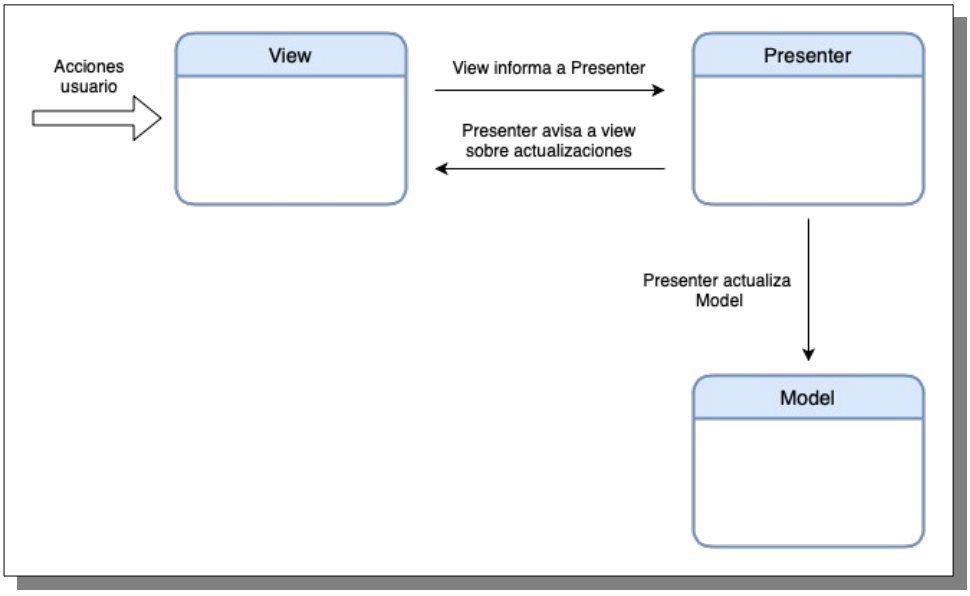

Figura 31: Arquitectura MVP

4. Implementación

4.1 Inicialización y configuración del proyecto

4.1.1 Creación del proyecto

Desde el entorno de desarrollo integrado Android Studio se ha creado un nuevo proyecto en el cual, como ya se ha adelantado en capítulos anteriores, se utilizará exclusivamente el lenguaje de programación Kotlin y donde la versión de Android mínima requerida es la 4.4.

Por otro lado, se ha decidido que la aplicación se utilizará exclusivamente en orientación vertical para evitar que la persona usuaria tenga que ir girando el dispositivo continuamente y porque se ha considerado que es la orientación que mejor se adapta a los contenidos que se pretenden presentar. Para conseguirlo se han especificado las propiedades configChanges y screenOrientation en cada una de las *activities* definidas en el archivo AndroidManifest.xml.

```
<activity
    android:name=".page.PageActivity"
    android:configChanges="orientation"
    android:screenOrientation="portrait" />
```

Figura 32: Establecer orientación vertical

4.1.2 Integración de Firebase en el proyecto

Para integrar el uso de la plataforma móvil en la nube Firebase se han seguido las instrucciones descritas en la guía *Agrega Firebase al proyecto de Android* (Firebase, 2020). La integración ha sido sencilla y no se han producido errores.

4.1.3 Configuración de Firebase Authentication

En este proyecto, se ha considerado que inicialmente es suficiente con proporcionar exclusivamente la autenticación mediante dirección de correo electrónico y contraseña. Para configurar e implementar este método de autenticación se han seguido las directrices presentadas en la guía *Autentica con Firebase mediante cuentas con contraseña en Android* (Firebase Authentication, 2020).

4.1.4 Creación de la Firebase Realtime Database

La base de datos en tiempo real se ha creado siguiendo los pasos descritos en la guía *Instalación y configuración en Android* (Firebase Realtime Database, 2020).

Una vez creada la base de datos se han establecido las reglas de seguridad. Concretamente, no se permite la escritura en ningún caso y la lectura solamente a usuarios autenticados.

4.1.5 Inicialización de Firebase Cloud Storage

Para inicializar el servicio de almacenamiento en la nube para que los usuarios de la aplicación puedan descargar los contenidos multimedia de cada una de las unidades didácticas, se han seguido las instrucciones de la guía *Comienza a usar Cloud Storage en Android* (Firebase Cloud Storage, 2020).

Una vez creado el *bucket* de almacenamiento se han definido las reglas de seguridad. Al igual que en la base de datos en tiempo real, no se permite la escritura en ningún caso y la lectura solamente a usuarios autenticados.

4.1.6 Librería RxJava/RxAndroid

Para simplificar la problemática que supone la ejecución de tareas concurrentes, asíncronas y en diferentes hilos de ejecución en la programación para dispositivos móviles, se ha decidido utilizar la librería de programación reactiva RxJava/RxAndroid (RxJava/RxAndroid, 2020).

El uso de esta librería es muy adecuado para ofrecer una buena experiencia de usuario, ya que permite que todas las operaciones de red como, por ejemplo, la obtención de datos remotos, se ejecuten de forma asíncrona en un hilo de ejecución en segundo plano y distinto al que ejecuta la interfaz de usuario, evitando experiencias de usuario no deseadas como bloqueos o congelaciones de pantalla.

Finalmente, comentar que RxJava/RxAndroid es una alternativa más adecuada que otras soluciones como AsyncTask, en las que se presentan problemas como fugas de memoria o de diferencias de funcionamiento entre las distintas versiones de Android.

4.1.7 Librería GlideToVectorForYou

Como se explicará más detalladamente en el apartado 4.5, en este proyecto se han utilizado exclusivamente gráficos vectoriales escalables (SVG).

En tiempo de desarrollo, cuando se requiere incorporar un gráfico vectorial escalable a un proyecto, se transforma en un elemento de diseño vectorial mediante la herramienta Vector Asset Studio, incluida en el entorno de desarrollo integrado Android Studio.

El problema es que en este proyecto se requiere poder visualizar gráficos vectoriales escalables descargados de Firebase Cloud Storage dinámicamente, en tiempo de ejecución. Por tanto, como no es posible utilizar la herramienta Vector Asset Studio ha sido necesaria la incorporación de la librería GlideToVectorForYou (GlideToVectorForYou, 2020) que sí lo permite.

4.1.8 Librería PhotoView

La librería PhotoView (PhotoView, 2020) se ha incorporado para proporcionar las funcionalidades de zoom y desplazamiento horizontal y vertical al visualizar un gráfico vectorial escalable.

4.1.9 Librería Material Drawer

La librería Material Drawer (Material Drawer, 2020) se ha incluido y utilizado para implementar el menú de la aplicación, ubicado en la pantalla de selección de nivel de alemán y al cuál se puede acceder mediante el botón hamburguesa, que se encuentra en el extremo superior izquierdo de la pantalla o mediante un desplazamiento con el dedo hacia la derecha iniciado desde el borde izquierdo de la misma.

4.2 Transformación del diagrama de clases en una estructura de nodos JSON

En la siguiente figura se presenta un ejemplo de la transformación inicial del diagrama de clases en una estructura de nodos JSON.


```
{
  "levels": [
    {
      "a2": {
        "titleDE": "Grundstufe II",
        "titleES": "Elemental",
        "didacticUnits": [
          {
            "titleDE": "Sich vorstellen",
            "titleES": "Presentarse",
            "status": "published",
            "statusMessage": "",
            "requiresResources": "false",
            "pages": [
              {
                "titleDE": "Sich vorstellen I",
                "titleES": "Presentarse I",
                "exerciseType": "correct_options",
                "statementDE": "Setze die Wörter richtig ein.",
                "statementES": "Completa con la palabra correcta.",
                "contents": [
                  {
                    "type": "",
                    "de": "Ich _____ Marie Müller.",
                    "imagePath": "",
                    "audioPath": "",
                    "correctOptionSequence": "1",
                    "options": [
                      {
                        "de": "heiße",
                        "imagePath": "",
                        "audioPath": ""
                      },
                      {
                        "de": "komme",
                        "imagePath": "",
                        "audioPath": ""
                      }
                    ]
                  }
                ]
              }
            ]
          }
        ]
      }
    }
  ]
}
```


Figura 33: Estructura de nodos JSON

4.2.1 Desnormalización del esquema JSON

Durante la implementación se ha detectado que para obtener la mejor relación de información requerida e información descargada, es preciso desnormalizar el esquema JSON (Firebase desnormalización, 2020). Concretamente, se han extraído a un nodo específico los índices de unidades didácticas pasando a un esquema con dos nodos raíz, a los que más adelante se ha sumado un tercero correspondiente a la lista de créditos de la aplicación. Mediante esta desnormalización es posible descargar los índices de unidades didácticas, o incluso solo uno de ellos, sin tener que descargar todo su contenido.

En las siguientes figura se puede observar el resultado de la desnormalización:

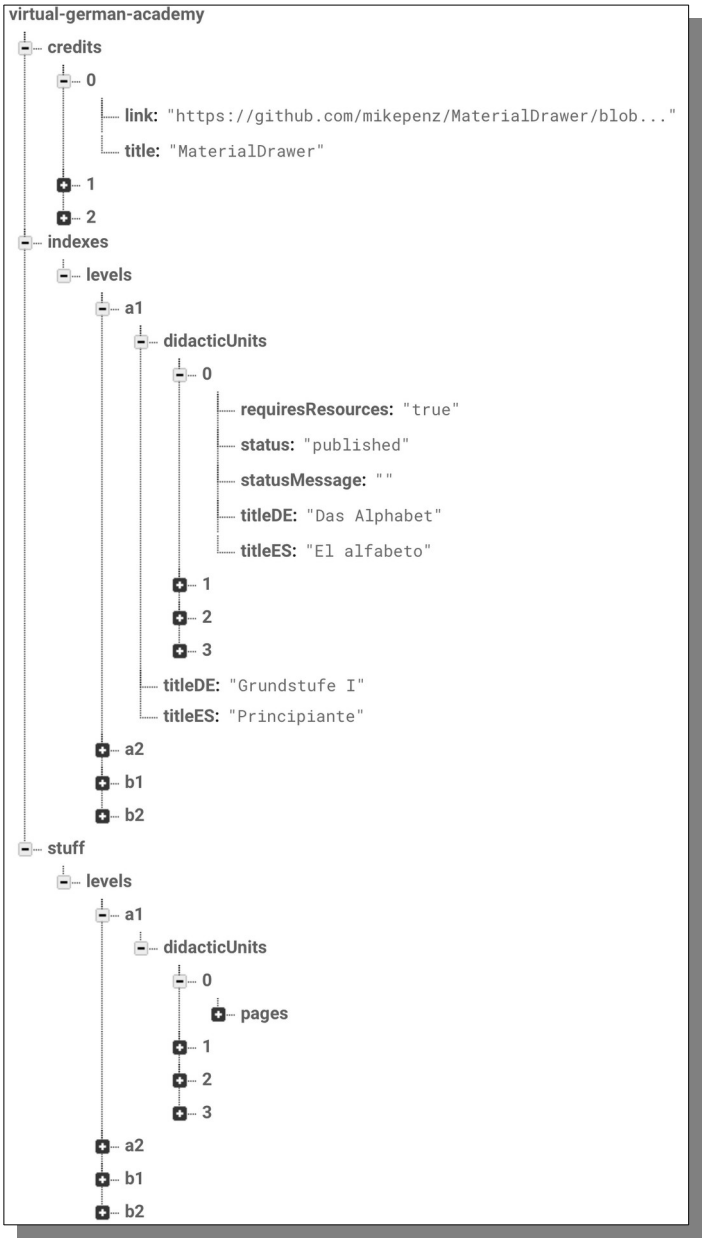

Figura 34: Estructura de nodos JSON desnormalizada

4.2.2 Integración de la estructura de nodos JSON en Firebase Realtime Database

En primer lugar han creado los datos correspondientes a los índices y contenidos de las unidades didácticas de los diferentes niveles, y la relación de créditos de la aplicación, respetando la definición del esquema JSON definido.

Para el ámbito de este trabajo los datos se han creado editando directamente el fichero JSON. En el futuro, para facilitar esta tarea se podría utilizar un editor especial de ficheros JSON o, aún mejor, diseñar e implementar una aplicación de backoffice específica y ajustada a la realización de esta tarea.

Una vez creados los datos en un fichero JSON, se han importado como Realtime Database desde la consola de Firebase.

4.3 Utilización de la aplicación sin conexión a Internet

Para hacer posible la utilización de la aplicación sin conexión a Internet, ha sido preciso configurar e implementar mecanismos de persistencia en disco de los datos requeridos por las distintas unidades didácticas.

La solución implementada permite utilizar sin conexión todos los contenidos a los que se haya accedido como mínimo una vez. En el futuro se podrían incluir otras opciones como, por ejemplo, la descarga personalizada de unidades didácticas o incluso de todas las unidades didácticas de un nivel de alemán.

4.3.1 Persistencia en disco de datos de Firebase Realtime Database

Los datos almacenados en la Firebase Realtime Database se sincronizan en tiempo real con cada cliente conectado y se mantienen disponibles mediante una caché cuando la aplicación no tiene conexión.

Para conseguir que los datos sigan estando disponibles incluso si el sistema operativo se reinicia, ha sido necesario habilitar la funcionalidad de persistencia en disco, de forma que la aplicación escribe los datos requeridos para mantener el estado en un directorio local del dispositivo.

La activación de la persistencia en disco ha sido sencilla, simplemente ha consistido en añadir la siguiente linea de código en la clase Application del paquete com.jbo.vga.main.

```
FirebaseDatabase.getInstance().setPersistenceEnabled(true)
```

En la guía *Habilita funciones sin conexión en Android* (Firebase funciones sin conexión, 2020), se describen los detalles del comportamiento de la persistencia en disco.

4.3.2 Persistencia en disco y sincronización de contenidos multimedia mediante Firebase Cloud Storage

En el caso de Firebase Cloud Storage, actualmente no se incluye una solución de persistencia en disco y sincronización automática, por lo que ha tenido que idear e implementar una solución propia.

Revisando la documentación de Firebase Cloud Storage (Firebase Cloud Storage, 2020) se ha visto que se tiene la posibilidad de descargar una lista de archivos de un directorio remoto y de descargar y consultar los metadatos de archivos. Por otro lado, se ha investigado cómo almacenar archivos en la caché local de la aplicación (Android developers local cache, 2020).

Teniendo en cuenta todo lo que se ha comentado en los párrafos anteriores, se ha implementado la siguiente solución de persistencia en disco y sincronización de contenidos multimedia:

1. En Firebase Cloud Storage se ha creado una estructura de directorios como la de la siguiente figura, para organizar los contenidos multimedia requeridos por las distintas unidades didácticas:

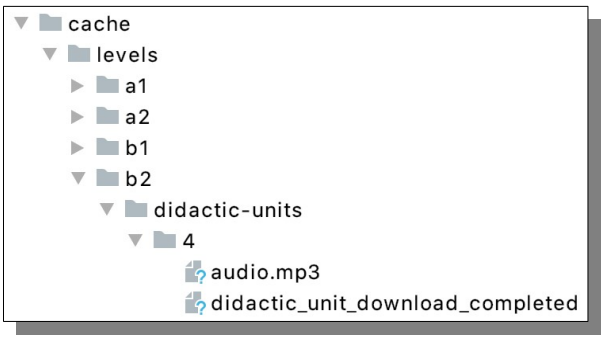

Figura 35: Estructura de directorios para organizar los contenidos multimedia

2. Cada vez que la persona usuaria de la aplicación selecciona una unidad didáctica que requiere recursos, se ejecuta el flujo presentado en la figura 36 para sincronizar los recursos multimedia remotos con los recursos multimedia locales.

3. Se intenta establecer conexión con Firebase Cloud Storage durante tres segundos. Este valor es configurable y se establece en la clase Application del paquete main mediante la siguiente sentencia:

```
FirebaseStorage.getInstance().maxDownloadRetryTimeMillis = 3000
```

4. Para determinar si la unidad didáctica en cuestión está disponible cuando no se tiene conexión se utiliza un archivo vacío denominado *didactic_unit_download_completed*, como marca de unidad didáctica descargada correctamente previamente.

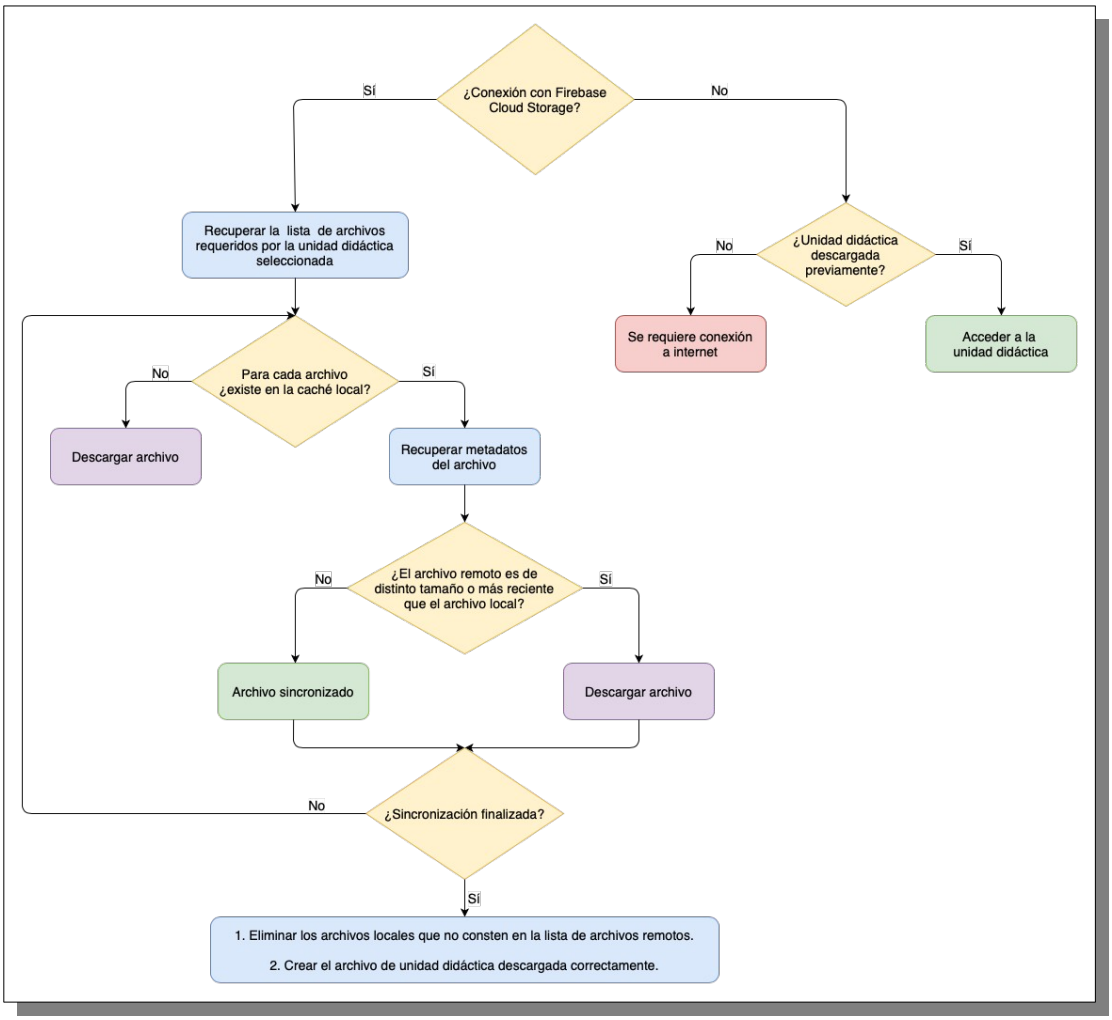

Figura 36: Sincronización de recursos multimedia de una unidad didáctica que requiere recursos

4.3.4 Consumo de espacio en disco de la aplicación

La persistencia en disco tanto de los datos recuperados de la Firebase Realtime Database como de Firebase Cloud Storage, provocan un incremento de consumo de espacio en disco de la aplicación.

Este consumo de espacio lo justifica la posibilidad de utilizar la aplicación sin conexión a Internet, que es uno de los objetivos de este trabajo, y el funcionamiento más fluido de la misma, ya que, por ejemplo, no se producirán cortes causados por problemas de conexión, al escuchar un audio. Se considera que si una persona realmente tiene interés en aprender alemán no tendrá inconveniente en que la aplicación consuma algo más de espacio. Concretamente, consumirá más o menos espacio, según la cantidad de unidades didácticas realizadas por la persona usuaria que, además, siempre tiene la posibilidad de liberar el espacio ocupado desde el apartado de gestión del espacio en la sección de aplicaciones del sistema operativo Android.

Finalmente, comentar que en el ámbito de este trabajo no se ha aplicado una gestión de espacio consumido, pero en el futuro se podría aplicar alguna estrategia de liberación de espacio como, por ejemplo, la eliminación de los archivos más antiguos. En cualquier caso, tal y como se indica en la documentación de Android Developers, el sistema operativo liberará espacio automáticamente si lo precisa.

4.4 Descarga de contenidos

En la aplicación existen los siguientes puntos de descarga y sincronización de contenidos:

Actividad home: como al autenticarse por primera vez a la aplicación es necesario disponer de conexión a Internet, se aprovecha para descargar de la Firebase Realtime Database y cargar en caché los índices de unidades didácticas de los niveles de alemán disponibles. Gracias a la desnormalización del esquema JSON descrita anteriormente, se descarga única y exclusivamente la información correspondiente a los índices de unidades didácticas. En futuros accesos, si no se dispone de conexión a Internet se visualiza la información que consta en caché.

Actividad índice: cada vez que se accede a un índice, si se dispone de conexión a Internet, se sincroniza con la información que consta en la Firebase Realtime Database. Si no se dispone de conexión a Internet se visualiza la información que consta en caché.

Actividad unidad didáctica: al seleccionar una unidad didáctica, por un lado se sincronizan los datos de la misma de Firebase Realtime Database y por otro sus recursos multimedia de Firebase Cloud Storage. Si no se dispone de conexión a Internet, solamente se accederá a la unidad didáctica seleccionada si consta una sincronización anterior de la misma. En caso contrario, se informará que se requiere conexión a Internet.

Actividad créditos: al acceder se sincroniza la relación de créditos que consta en la Firebase Realtime Database. Si no se dispone de conexión a Internet se visualiza la información que consta en caché.

4.5 Utilización de gráficos vectoriales escalables (SVG)

Se ha decidido que todos los recursos gráficos utilizados en este proyecto sean gráficos vectoriales escalables porque ofrecen la ventaja de poder ser redimensionados sin pérdida de calidad, lo cuál es fundamental para la correcta visualización de la aplicación sobre dispositivos con distintos tamaños de pantalla.

En la versión actual del proyecto se incluyen tres modalidades de recursos gráficos:

Cargados en tiempo de desarrollo: el archivo SVG es importado y convertido en un elemento de diseño vectorial (archivo XML) mediante la herramienta Vector Asset Studio, incluida en el entorno de desarrollo integrado Android Studio (Android Developers gráficos vectoriales, 2020). Una vez convertido está disponible en forma de recurso *drawable*.

Cargados en tiempo de ejecución estáticos: el archivo SVG es cargado desde la caché de la aplicación, ubicada en el disco local, mediante la librería GlideToVectorForYou. El tamaño de visualización final del gráfico es proporcional a las dimensiones de la pantalla.

Cargados en tiempo de ejecución aumentables: el archivo SVG también es cargado desde la caché de la aplicación, ubicada en el disco local, mediante la librería GlideToVectorForYou pero, en este caso, se combina con las funcionalidades de zoom y desplazamiento ofrecidas por la librería PhotoView. Esta modalidad es ideal para la presentación de las páginas de teoría, ya que se consigue que sean redimensionables sin pérdida de calidad.

Finalmente, comentar que durante el proceso de investigación, se ha visto que mediante otras librerías como, por ejemplo Glide (Glide, 2020), es posible descargar y visualizar el archivo SVG directamente en un componente ImageView a través de la referencia al archivo proporcionada por Firebase Cloud Storage. Lamentablemente, no es sido una solución válida para este proyecto, ya que uno de los objetivos del mismo es poder utilizar la aplicación sin conexión a Internet.

4.6 Estructura del proyecto

En la siguiente figura se presenta la estructura del proyecto en Android Studio:

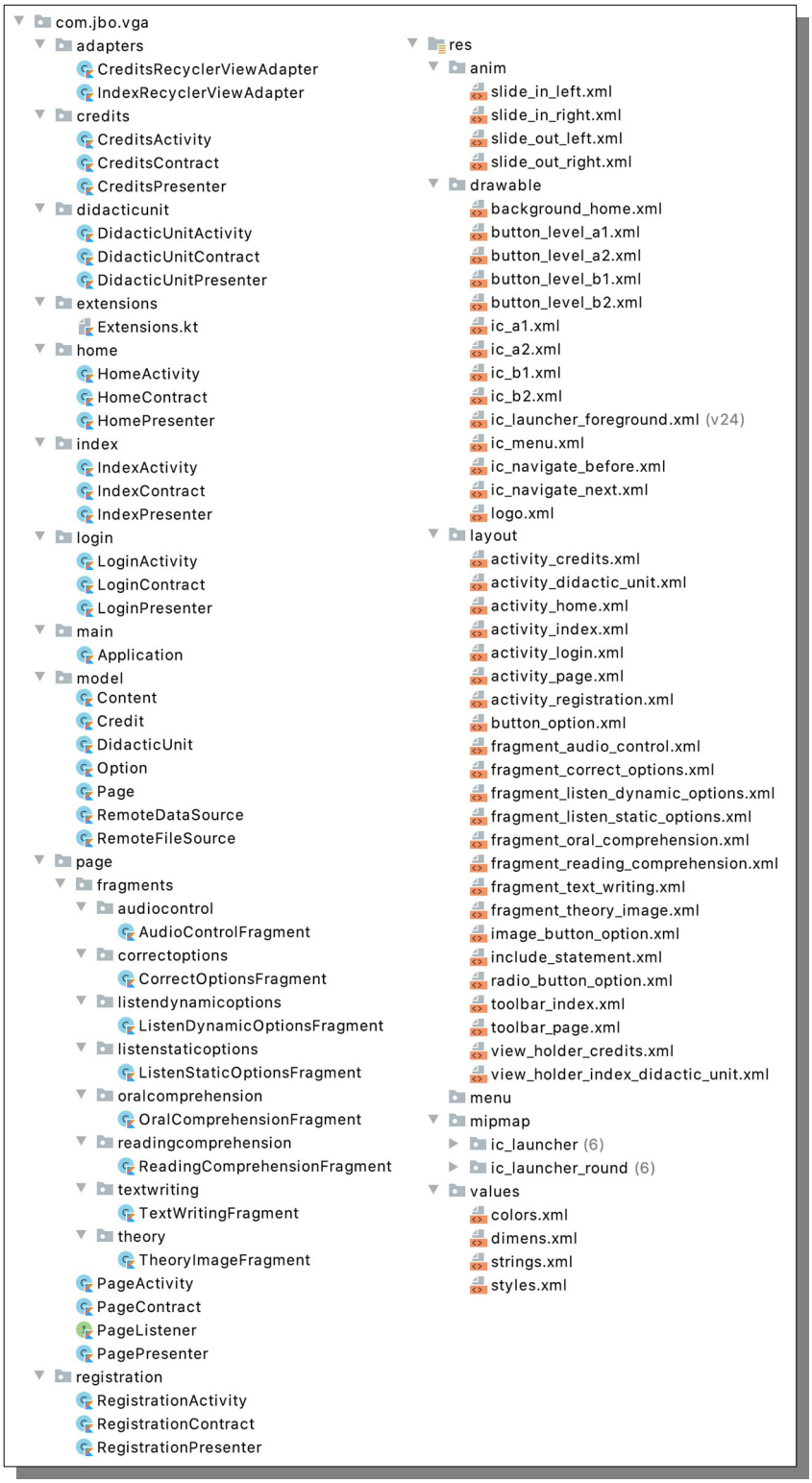

Figura 37: Estructura del proyecto en Android Studio

Como se comentó en el apartado 3.2.2 se ha decidido utilizar el patrón de diseño Model View Presenter (MVP). Se ha implementado realizando una división en paquetes de funcionalidades compuestos por una actividad, un presentador y un contrato, que mediante la definición de interfaces hace posible el desacople de la vista y el presentador. Véanse los paquetes credits, didacticunit, home, index, login, page y registration.

A continuación una breve descripción del resto de paquetes que componen el proyecto:

adapters: contiene los adaptadores utilizados para las *recycler views* de créditos y de índice de unidades didácticas.

main: contiene inicializaciones generales de la aplicación como la activación de la persistencia en disco de Firebase Realtime Database, y el tiempo máximo de reintento de descarga de Firebase Cloud Storage.

extensions: se incluye una librería de extensiones y funciones generales, utilizadas desde diversos puntos del proyecto.

model: contiene las clases necesarias para implementar el modelo de la aplicación. Destacar las clases RemoteDataSource y RemoteFileSource que hacen posible la comunicación con Firebase Realtime Database y Firebase Cloud Storage respectivamente.

4.6.1 Reutilización: layout includes, layouts de componentes, adapter parametrizado y fragmentos

En este apartado se quiere poner de manifiesto que siempre que ha sido posible se ha optado por la reutilización de código. Claros ejemplos de ello son:

Include de la barra de herramientas índice: todos los índices de unidades didácticas utilizan la misma barra de herramientas.

Include de la barra de herramientas página: todas las páginas utilizan la misma barra de herramientas.

Include del enunciado del ejercicio: inclusión del enunciado en todas las páginas.

Layouts de Buttons, ImageButtons y RadioButtons: en algunos tipos de ejercicio es necesario crear estos componentes dinámicamente. Se han creado layouts con las características generales a las que se les añaden en tiempo de ejecución las características específicas como, por ejemplo, el color condicionado por el nivel de idioma.

Definición de un solo adapter parametrizado: utilizado para todos los índices de unidades didácticas.

Uso de fragmentos: se ha creado un fragmento con layout propio para cada tipo de ejercicio y el fragmento AudioControlFragment, también con layout propio, que permite controlar la reproducción de un fichero de audio. Según el

tipo de ejercicio seleccionado por la persona usuaria, se incorporan uno o más fragmentos a la página.

4.6.2 Comunicación entre la actividad Page y los fragmentos de tipo de ejercicio

Existen diversas formas de invocar funcionalidades de la actividad en los fragmentos pero, actualmente, según Android Developers (Android Developers Activity-Fragment comunication, 2020) la más adecuada es a través de una interfaz. Por ello, en el proyecto se ha creado la interfaz PageListener que contiene la definición de los métodos que pueden ser invocados desde los distintos fragmentos.

```
interface PageListener {
    fun getImageUri(resourcePath: String): Uri
    fun getAudioUri(resourcePath: String): Uri
    fun getFeedbackEmailSubject(level: String): String
    fun loadSvgImage(uri: Uri, imageView: ImageView)
    fun addAudioFragment(level: String, audioPath: String)
    fun onHideSoftKeyboard(view: View)
    fun onContentFinished()
}
```

Figura 38: Interfaz PageListener

4.7 Envío de ejercicios mediante correo electrónico para obtener *feedback* correctivo

En los ejercicios donde se permite su envío para que sean corregidos por profesorado nativo, al pulsar sobre el correspondiente botón se inicia un *intent* que permite seleccionar el gestor de correo electrónico preferido de la persona usuaria y al cuál se adjunta la dirección del destinatario, el asunto y el contenido del mensaje, de forma que el mensaje es compuesto automáticamente y la persona usuaria solamente tiene que revisarlo y enviarlo.

4.8 Animaciones de paginación de un libro

Durante el proceso de implementación ha surgido la idea de incluir las animaciones de traslación a izquierda y derecha al cambiar de página, regresar al índice de unidades didácticas, etc. De este modo se ha conseguido que al pasar de una página a otra se tenga la sensación de estar pasando las páginas de un libro. Para conseguirlo se han definido e integrado las animaciones ubicadas en el directorio de recursos anim.

4.9 Sección de créditos

Tal y como se ha descrito en el apartado 4.1 de este capítulo, para llevar a cabo la implementación de este proyecto de ha hecho uso diversas librerías de terceros con licencia Apache versión 2.0. Para cumplir con las condiciones de esta licencia se ha creado la sección de créditos, compuesta por una lista de las librerías utilizadas con un vínculo a su licencia y a la que se puede acceder desde el menú de la pantalla de selección de nivel de alemán.

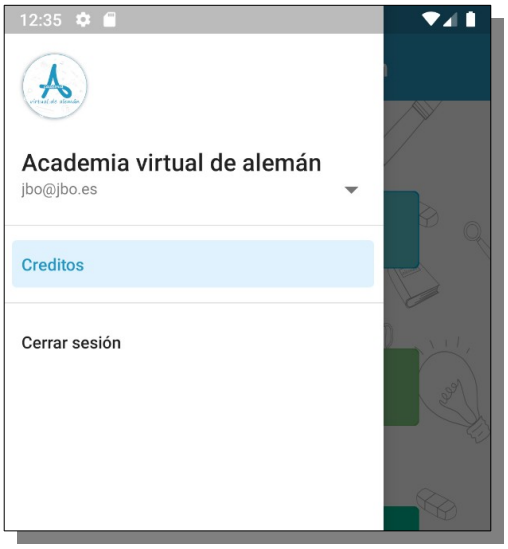

Figura 39: Menú de acceso a la sección de créditos

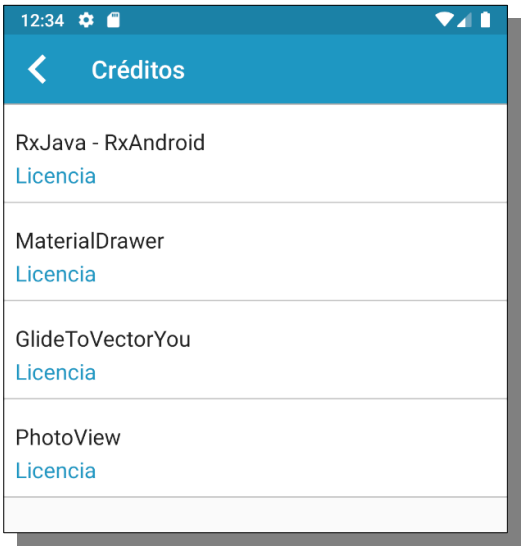

Figura 40: Sección de créditos

4.10 Herramientas Lint y Android Profiler

A medida que se ha avanzando en la implementación del proyecto, se han utilizado las herramientas incluidas en el entorno de desarrollo integrado Android Studio, Lint (Lint, 2020) y Android Profiler (Android Profiler, 2020).

Lint permite analizar el código con el objetivo de localizar código poco estructurado, uso de elementos que ya no están disponibles, llamadas de API no admitidas por versiones de API de destino, etc. Se han corregido todos los problemas que ha sido posible.

Respecto a las herramientas de Android Profiler, proporcionan datos en tiempo real que ayudan a analizar el uso de los recursos de la CPU, la memoria, la red y la batería. En las inspecciones llevadas a cabo no se ha detectado ningún uso inapropiado de recursos.

5. Pruebas

5.1 Creación de tests unitarios

En el patrón de diseño Model View Presenter (MVP) es posible realizar tests unitarios del presentador porque:

1. Las vistas extienden de una Activity o de un Fragment y, tanto el modelo como el presentador, no extienden de ninguna clase específica del framework de Android.
2. En el presentador no se hace referencia a clases del framework de Android, como Context, View o Intent.
3. La comunicación entre el presentador y la vista se lleva a cabo a través de las interfaces definidas en las clases contrato, haciendo posible su desacople y el mock de la interfaz de la vista en el presentador.

Con el tiempo disponible no ha sido posible implementar tests de nivel intermedio o incluso superior. Los tests de nivel inferior implementados mediante el framework Mockito (Mockito, 2020) constituyen la base del conjunto de pruebas de la aplicación, por lo que si todos los tests unitarios funcionan correctamente es muy probable que en conjunto funcionen correctamente. Concretamente, permiten validar la funcionalidad y los contratos de todas las clases de la aplicación y tienen la ventaja de no requerir emulador o dispositivo físico para su ejecución, por lo que son muy rápidos.

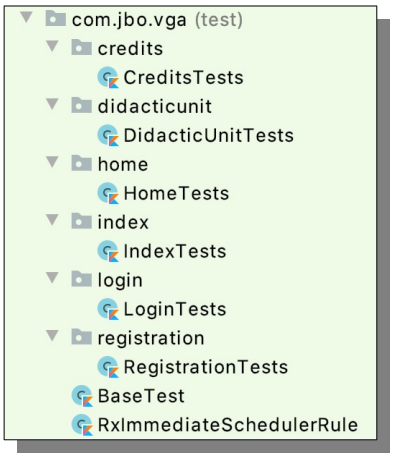

Figura 41: Tests unitarios

5.2 Pruebas funcionales

Las pruebas funcionales han consistido en la verificación del correcto funcionamiento de todos los casos de uso definidos en el apartado 3.1 primero sobre el emulador y a continuación sobre los dispositivos físicos descritos en el apartado 1.4.1 Entorno de desarrollo.

Al realizar las pruebas sobre el dispositivo Motorola Moto G, con versión de Android 4.4.4, se ha producido un error grave en la carga de recursos gráficos vectoriales escalables (archivos SVG) referenciados directamente en un layout. Para solucionarlo se ha tenido que aplicar la solución de retrocompatibilidad de elementos de diseño vectorial (Android Developers vector compatibility, 2020).

Por otro lado, sobre el dispositivo Samsung Galaxy M21 con versión de Android 10, se ha detectado un problema de visualización de las imágenes del ejercicio escuchar opciones estáticas: el espacio horizontal no era suficiente para soportar la anchura calculada de las opciones. El error se ha producido porque la relación de aspecto de la pantalla de este dispositivo difiere de la de todos los demás. Concretamente, es más alta que ancha. Se ha solucionado mejorando el algoritmo de creación de las opciones estáticas, estableciendo un porcentaje de anchura de la pantalla para cada una de ellas.

5.3 Pruebas por parte de personas usuarias colaboradoras

Se ha solicitado a las mismas personas usuarias que validaron el prototipo de alta fidelidad que verifiquen el correcto funcionamiento de la aplicación.

Destacar la importancia de estas pruebas porque después de llevar a cabo los tests unitarios y las pruebas funcionales, se podría pensar que es muy poco probable que reste algún problema por resolver, pero la realidad es que se han detectado tres:

1. En el ejercicio de escuchar opciones dinámicas, el botón stop, inicialmente no estaba deshabilitado. No era un problema grave pero no era correcto. Se ha solucionado.
2. Error NullPointerException al acceder a una unidad didáctica y volver atrás dos veces seguidas. Era un error muy grave porque abortaba la ejecución de la aplicación. Se ha revisado y se ha visto que es un error conocido relacionado con el uso de la librería RxJava/RxAndroid (RxJava/RxAndroid Error handling, 2020). Concretamente se produce una interrupción por cancelación y consecuentemente no es posible emitir ningún valor. Se ha solucionado comprobando que el emisor no haya sido eliminando antes de emitir.
3. En los índices de unidades didácticas, al pulsar más de una vez sobre una de ellas, se abrían tantas instancias como veces se había pulsado. Se ha solucionado introduciendo un mecanismo que asegura que solo se gestione la primera pulsación.

6. Evaluación de la evolución del trabajo

6.1 Hito: entrega de la PEC2 – Diseño

Gestión del *feedback* obtenido de la entrega de la PEC1

1. Incorporación a la bibliografía de los enlaces a los mercados de aplicaciones presentadas en el estudio de mercado.

Cumplimiento de la planificación del punto 1.4.2 Planificación

1. Diseño conceptual: se han requerido 12 horas en lugar de 8 principalmente por la cantidad de tiempo que ha sido necesario para la creación y maquetación de las fichas de las personas y escenarios.
2. Prototipado: han sido suficientes 17 horas en lugar de 21, recuperando las 4 horas extras requeridas en el diseño conceptual.

Otras actuaciones

1. Inclusión al proyecto del servicio Firebase Cloud Storage que permitirá almacenar y distribuir el contenido multimedia requerido en las unidades didácticas.
2. Adición del objetivo opcional OO07 – Adición de más idiomas base, como el inglés, francés, etc.

6.2 Hito: entrega de la PEC3 – Implementación

Gestión del *feedback* obtenido de la entrega de la PEC2

1. Incorporación a la memoria del anexo “Evaluación de la evolución del trabajo”, incluyendo un resumen del informe presentado al entregar la PEC2 y los cambios introducidos desde entonces.
2. Eliminación del apartado 1.6 *Breve descripción de los otros capítulos de la memoria*. Hubo un malentendido por parte del alumno, ya que en lugar de una previsión de los capítulos al entregar la PEC1 creyó que lo que se requería era un apartado con el resumen de los capítulos que componen la memoria, como es habitual en muchos libros.

Cumplimiento de la planificación del punto 1.4.2 Planificación

1. La planificación del desarrollo de las unidades didácticas del nivel A1 de alemán se ha visto seriamente alterada a causa de la necesidad de implementación de una solución de caché y de sincronización de recursos multimedia entre el dispositivo local y el servicio de almacenamiento en la nube Firebase Cloud Storage. El motivo del incumplimiento de esta parte de la planificación ha sido el desconocimiento del funcionamiento de Firebase Cloud Storage, ya que no se había utilizado nunca hasta el momento. Al incluirlo al proyecto, no se detectó el riesgo de que el servicio no incorpore una solución de caché y sincronización de archivos, como sí ocurre en Firebase Realtime Database.

Afortunadamente, tras implementar las unidades didácticas del nivel de alemán A1, el desarrollo de las unidades didácticas de los niveles restantes ha requerido menos tiempo del planificado, lo que en combinación a la realización de 8 horas en lugar de 4 el sábado 7 y el domingo 8 de noviembre ha permitido reengancharse a la planificación.

2. En la planificación no se reservó tiempo específico para la redacción de los capítulos de implementación y pruebas de esta memoria. Se han ido realizando anotaciones durante todo el proceso de implementación y pruebas y durante los días del 6 a 8 de diciembre (fin de semana y dos días festivos) se han realizado 12 horas extraordinarias para conseguir redactar esta documentación.

Otras actuaciones

1. Adición de los nuevos atributos status y statusMessage en la clase DidacticUnit, que permiten especificar el estado de publicación de una unidad didáctica y una descripción del mismo. Por ejemplo, se puede indicar que la unidad didáctica “Los colores” estará disponible a partir del 1 de febrero de 2021.

2. Adición del atributo requiresResources en la clase didacticUnit que permite especificar si una unidad didáctica requiere recursos, de modo que solo se invocará la llamada al Firebase Cloud Storage si el valor del atributo es verdadero.

3. Creación de un repositorio Git privado para registrar la evolución del proyecto y poder recuperar enfoques o soluciones descartadas.

6.3 Hito: entrega de la PEC4 – Entrega final

Gestión del *feedback* obtenido de la entrega de la PEC3

1. No ha surgido ninguna acción o mejora a realizar.

Cumplimiento de la planificación del punto 1.4.2 Planificación

1. La planificación se ha cumplido sin contratiempos. Solamente comentar que el hecho de disponer de unos días de vacaciones antes de Navidad ha permitido adelantar las horas programadas del 27 al 30 de diciembre y consecuentemente anticipar la entrega de la PEC4 al día 26 de diciembre.

Otras actuaciones

1. Corrección en la composición del texto de *feedback* de las dos cuestiones a responder por escrito del ejercicio de compresión oral del nivel B2 de alemán. A causa de una modificación previa se estaba utilizando un campo de texto incorrecto para componer la respuesta a enviar mediante correo electrónico.

2. Se ha detectado que al cambiar de app el audio se seguía reproduciendo. Se ha solucionado pausando la reproducción en el método de ciclo de vida *onPause* del fragmento AudioControlFragment.

3. Se ha decidido liberar el código fuente bajo la licencia Apache versión 2.0. Para ello se ha añadido un aviso en las clases más importantes del desarrollo y se ha incorporado al proyecto el archivo LICENSE.txt, que contiene una copia de la licencia Apache 2.0.
4. Adición del apartado 4.10 Herramientas Lint y Android Profiler.

7. Conclusiones

La realización de este trabajo ha sido una experiencia muy enriquecedora en la que, además de poder aplicar gran parte de los conocimientos adquiridos en el máster, he tenido que profundizar mis conocimientos en las fases de análisis, planificación, diseño e implementación de una aplicación móvil. A continuación mis conclusiones de cada una de estas fases:

Análisis: ha sido muy laboriosa, especialmente el estudio de mercado, que ha requerido investigar y procesar gran cantidad de información.

Planificación: ha resultado muy interesante evaluar la evolución de la planificación establecida y tener que tomar decisiones ante situaciones imprevistas que hacían peligrar su cumplimiento. He aprendido que, además de definir un tiempo adicional para cada hito del proyecto para poder reaccionar ante imprevistos, es importante definir una bolsa de tiempo adicional para imprevistos globales.

Diseño: en mi opinión es clave ya que un buen diseño proporciona un margen de reacción ante problemas y situaciones imprevistas en la fase de implementación. El enfoque de diseño centrado en el usuario (DCU) ha facilitado el trabajo y ha esclarecido las ideas. Me han sorprendido las diferentes visiones que han surgido de la puesta en común y evaluación del prototipo, y lo efectiva que puede resultar esta práctica para evitar modificaciones de requisitos en la fase de implementación.

Implementación: de esta fase destacar el uso de programación reactiva mediante la librería RxJava/RxAndroid, la desnormalización del esquema JSON, el uso de gráficos vectoriales escalables (SVG), la persistencia en disco de datos descargados de Firebase Realtime Database, la persistencia en disco y sincronización de recursos multimedia mediante Firebase Cloud Storage y la reutilización y generalización de código.

Estoy bastante satisfecho con el trabajo realizado en cada una de las fases. Como nota negativa, manifestar que en la fase de diseño no era consciente de que el servicio de almacenamiento en la nube Firebase Cloud Storage actualmente no incluye una solución de persistencia en disco y sincronización automática. En la fase de implementación me topé con este imprevisto y tuve que desarrollar una solución propia, lo cuál hizo peligrar el cumplimiento de la planificación prevista. Afortunadamente, como se ha acertado con el resto del diseño ha sido posible asumir el imprevisto y reengancharse a la planificación. He aprendido la lección de analizar aún más profundamente si los productos y servicios que se pretenden utilizar proporcionan las funcionalidades requeridas.

Respecto al logro de los objetivos planteados inicialmente, comentar que:

1. Los objetivos principales se han cumplido, aunque para cumplir estrictamente con los que están relacionados directamente con el aprendizaje, como el *OP03 - Proporcionar un método de aprendizaje como si se tratase de una persona nativa* y el *OP04 - Preparar al alumnado para los exámenes oficiales de alemán de los niveles A1, A2, B1 y B2*, queda pendiente la incorporación de más unidades didácticas e incluso de nuevos tipos de ejercicio.

   Destacar que lo que se ha conseguido en este trabajo es una infraestructura que permite presentar contenido de una forma organizada, además de posibilitar la interacción entre alumnado y profesorado nativo. Actualmente, es posible crear la cantidad de ejercicios que se quiera de los siete tipos definidos, por ejemplo, del tipo de ejercicio consistente en escribir un texto sobre un tema que además puede ser enviado para ser corregido, se pueden incorporar tantos como hagan falta, simplemente creando el contenido requerido en la base de datos.

   Finalmente, remarcar que la aplicación desarrollada en este trabajo se puede adaptar a la enseñanza de prácticamente cualquier otra temática como las matemáticas, la geografía, la historia, la preparación de oposiciones, la teoría del carnet de conducir, etc. para las cuales también puede ser muy útil la presentación de contenidos de forma organizada y fomentar la interacción entre alumnado y profesorado. En cualquier caso, la clave está en la generación de contenidos adecuados y de calidad.

2. Los objetivos funcionales y los no funcionales se han cumplido sin nada que objetar. Solamente manifestar que el objetivo no funcional más costoso de cumplir ha sido el *RNF03 - La aplicación se podrá utilizar sin conexión*, ya que ha sido un poco como remar contracorriente porque en general, el entorno, las librerías, los servicios, etc. están más bien orientados al funcionamiento con conexión.

3. Los objetivos opcionales no ha sido posible abordarlos.

En cualquier caso, el balance es positivo porque se ha conseguido implementar prácticamente todo lo que se había planificado.

Como objetivo y reto personal, estoy muy satisfecho por haber conseguido implementar todo el proyecto utilizando exclusivamente el lenguaje de programación Kotlin, que como se ha justificado en el apartado 1.3.2 apunta a ser el lenguaje de programación del futuro para el desarrollo de aplicaciones móviles en Android. Lo que más me ha gustado de Kotlin es su codificación corta, concisa y directa.

Finalmente, una lista de posibles trabajos futuros a realizar además de los objetivos opcionales que no ha sido posible abordar en este trabajo:

1. Creación de una aplicación de backoffice que permita gestionar la creación y publicación de contenidos.
2. Mejorar el trabajo sin conexión, por ejemplo, permitiendo la descarga al disco local de los datos y recursos multimedia de todo un nivel de alemán.
3. Mejorar el fragmento de control de audio de forma que incluya una barra de desplazamiento que permita avanzar o retroceder la reproducción.
4. Incorporar una estrategia o política de liberación de memoria caché consumida por la aplicación.
5. Crear nuevos tipos de ejercicio, como por ejemplo el de expresión oral, que permita realizar una grabación y enviarla al profesorado nativo para su corrección.
6. En los ejercicios de redacción de texto, almacenar en caché local la respuesta redactada y recuperarla al volver a acceder al ejercicio. Se tiene que tener en cuenta que la aplicación puede ser utilizada por distintas personas usuarias sobre el mismo dispositivo móvil.
7. Incorporar una sección que permita el envío de dudas al profesorado nativo.
8. Incorporar una sección que permita enviar textos de temática libre para su corrección y obtención de *feedback* correctivo.
9. En los ejercicios de redacción de textos, visualizar la cantidad de palabras escritas y las que restan para cumplir con el número de palabras especificado en el enunciado del ejercicio.
10. Creación de contenido personalizado para el alumnado, que podría aparecer, por ejemplo, como una unidad didáctica de actividades de refuerzo personales.

8. Glosario

APP Abreviatura de la palabra en inglés *application* (aplicación). Se refiere sobre todo a aplicaciones destinadas a dispositivos móviles.

DCU Diseño centrado en el usuario. Enfoque de diseño cuyo proceso está dirigido por información sobre las personas que van a hacer uso del producto.

JSON JavaScript Object Notation. Formato ligero de intercambio de datos, que es sencillo de leer y escribir para humanos, y simple de interpretar y generar para las máquinas.

NoSQL Sistema gestor de base de datos donde no se utiliza el lenguaje SQL y en el cual se almacena información no estructurada, es decir, no existen tablas y columnas como en el modelo relacional.

SVG Gráficos vectoriales escalables cuya principal ventaja es que no se produce pérdida de calidad al ser redimensionados.

UML Unified Modeling Language. Estándar internacional para crear esquemas, diagramas y documentación relativa los desarrollos de software.

5G Quinta generación de las tecnologías y estándares de comunicación inalámbrica.

9. Bibliografía

ANDROID DEVELOPERS ACTIVITY-FRAGMENT COMUNICATION (2020). *Como comunicar contenido con otros fragmentos*, consultada el 19 de noviembre de 2020.
<https://developer.android.com/training/basics/fragments/communicating>

ANDROID DEVELOPERS GRÁFICOS VECTORIALES (2020). *Cómo agregar gráficos vectoriales de varias densidades*, consultada el 14 de noviembre de 2020.
<https://developer.android.com/studio/write/vector-asset-studio>

ANDROID DEVELOPERS LOCAL CACHE (2020). *Cómo acceder a archivos específicos de la app*, consultada el 1 de noviembre de 2020.
<https://developer.android.com/training/data-storage/app-specific?hl=es-419>

ANDROID DEVELOPERS VECTOR COMPATIBILITY (2020). *Solución de retrocompatibilidad de elementos de diseño vectoriales*, consultada el 6 de diciembre de 2020.
<https://developer.android.com/guide/topics/graphics/vector-drawable-resources#vector-drawables-backward-solution>

ANDROID PROFILER (2020). *Como medir el rendimiento de las apps con Android Profiler*, consultada el 15 de diciembre de 2020.
<https://developer.android.com/studio/profile/android-profiler?hl=es-419>

APRENDER ALEMÁN (2020). *Aprender alemán*, consultada el 20 de septiembre de 2020.
Android:
<https://play.google.com/store/apps/details?id=pl.rosmedia.rozmowkihisger>
iOS:
<https://apps.apple.com/ve/app/aprender-alemán-audio-curso-y-vocabulario-gratis/id402727185>

BABBEL (2020). *Babbel*, consultada el 20 de septiembre de 2020.
Android:
<https://play.google.com/store/apps/details?id=com.babbel.mobile.android.en>
iOS:
<https://apps.apple.com/es/app/babbel-aprender-idiomas/id829587759>

BUSUU (2020). *Busuu*, consultada el 20 de septiembre de 2020.
Android:
<https://play.google.com/store/apps/details?id=com.busuu.android.enc>
iOS:
<https://apps.apple.com/us/app/busuu-fast-language-learning/id379968583>

DIAGRAMS.NET (2020). *Free online diagram software for making flowcharts, process diagrams, org charts, UML, etc.* Consultada el 17 de octubre de 2020. <https://app.diagrams.net>

DUOLINGO (2020). *Duolingo*, consultada el 20 de septiembre de 2020.
Android:
<https://play.google.com/store/apps/details?id=com.duolingo>
iOS:
<https://apps.apple.com/es/app/duolingo/id570060128>

FIREBASE (2020). *Agrega Firebase al proyecto de Android*, consultada el 31 de octubre de 2020.
<https://firebase.google.com/docs/android/setup?hl=es#assistant>

FIREBASE AUTHENTICATION (2020). *Autentica con Firebase mediante cuentas con contraseña en Android*, consultada el 31 de octubre de 2020.
<https://firebase.google.com/docs/auth/android/password-auth?hl=es>

FIREBASE CLOUD STORAGE (2020). *Comienza a usar Cloud Storage en Android*, consultada el 1 de noviembre de 2020.
<https://firebase.google.com/docs/storage/android/start>

FIREBASE DESNORMALIZACIÓN (2020). *Estructura tu base de datos*, consultada el 3 de noviembre de 2020.
<https://firebase.google.com/docs/database/web/structure-data?hl=es>

FIREBASE FUNCIONES SIN CONEXIÓN (2020). *Habilita funciones sin conexión en Android*, consultada el 1 de noviembre de 2020.
<https://firebase.google.com/docs/database/android/offline-capabilities?hl=es>

FIREBASE REALTIME DATABASE (2020). *Instalación y configuración en Android*, consultada el 2 de noviembre de 2020.
<https://firebase.google.com/docs/database/android/start?hl=es>

FLAMARICH ZAMPALO, JORDI (2017). *Investigación y requisitos de usuario. Conceptualización*. PID-00245395. FUOC.

GLIDE (2020). *Glide Tutorial For Android*, consultada el 14 de noviembre de 2020.
<https://www.raywenderlich.com/2945946-glide-tutorial-for-android-getting-started>

GLIDETOVECTORFORYOU (2020). *Load SVGs with Glide*, consultada el 10 de noviembre de 2020.
<https://github.com/corouteam/GlideToVectorYou>

JAN SEMLER, KIRA TSCHIERSCHKE (2020). *App-Design: Das umfassende Handbuch. Alles zur Gestaltung, Usability und User Experience von iOS-, Android- und Web-Apps*. Rheinwerk Design.

JUSTINMIND (2020). *All-in-one prototyping tool for web and mobile apps*, consultada el 16 de octubre de 2020.
<https://www.justinmind.com>

LEARNMATCH (2020). *Aprender idiomas, aprender inglés*, consultada el 20 de septiembre de 2020.
Android:
<https://play.google.com/store/apps/details?id=net.learnmatch.app>
iOS:
<https://apps.apple.com/ar/app/learnmatch-aprender-idiomas/id1178670975>

LINT (2020). C*ómo mejorar tu código con comprobaciones de lint*, consultada el 15 de diciembre de 2020.
<https://developer.android.com/studio/write/lint?hl=es-419>

MONDLY (2020). *Aprende idiomas gratis – Mondly*, consultada el 20 de septiembre de 2020.
Android:
<https://play.google.com/store/apps/details?id=com.atistudios.mondly.languages>
iOS:
<https://apps.apple.com/es/app/mondly-aprender-33-idiomas/id987873536>

MATERIAL DRAWER (2020). *The flexible, easy to use, all in one drawer library for you Android project*, consultada el 25 de noviembre de 2020.
<https://github.com/mikepenz/MaterialDrawer>

MOCKITO (2020). *Tasty mocking framework for unit tests*, consultado el 5 de diciembre de 2020.
<https://site.mockito.org>

MSUTODAY (2020). *How effective are language learning apps?* Artículo del 23 de abril de 2020, consultado el 17 de septiembre de 2020.
<https://msutoday.msu.edu/news/2020/how-effective-are-language-learning-apps/>

PHOTOVIEW (2020). *PhotoView aims to help produce an easily usable implementation of a zooming Android ImageView*, consultada el 12 de noviembre de 2020.
<https://github.com/chrisbanes/PhotoView>

QUÉIDIOMA (2020). *¿Qué idiomas se hablan en Europa?* Información consultada el 22 de septiembre de 2020.
<https://queidioma.com/hablan-en-europa/#_Aleman>

REVIEW 42 (2020). *39+ Smartphone Statistics You Should Know in 2020*. Artículo del 5 de junio de 2020, consultado el 28 de septiembre de 2020.
<https://review42.com/smartphone-statistics/>

RXJAVA/RXANDROID (2020). *Reactive extensions for Android*, consultada el 6 de noviembre de 2020.
<https://github.com/ReactiveX/RxAndroid>

RXJAVA/RXANDROID ERROR HANDLING (2020). *What's different in 2.0 – Error handling*, consultada el 7 de diciembre de 2020.
<https://github.com/ReactiveX/RxJava/wiki/What's-different-in-2.0#error-handling>

SKETCHIZE (2020). *Start to convert your ideas into lovely sketches with Sketchize*, consultada el 12 de octubre de 2020.
<https://www.sketchize.com>

STATCOUNTER (2020). *Mobile Operating System Market Share Worldwide*. Estadística que engloba el período de septiembre de 2019 a septiembre de 2020, consultada el 3 de octubre de 2020.
<https://www.statista.com/statistics/640059/market-share-mobile-operating-systems-spain/>

STATISTA (2020). *Market share of mobile operating systems in Spain from 2010 to 2019*. Estadística de 2010 a 2020 publicada en febrero de 2020, consultada el 3 de octubre de 2020.
<https://gs.statcounter.com/os-market-share/mobile/worldwide>

THOMAS THEIS (2019). *Einstieg in Kotlin: Apps entwickeln mit Android Studio. Keine Vorkenntnisse.* Rheinwerk Computing.

TRAVEL STUDENT (2020). *Global Trends in foreign language demand and proficiency.* Artículo de 2020, consultado el 17 de septiembre de 2020.
<https://studenttravelplanningguide.com/global-trends-in-foreign-language-demand-and-proficiency/>

UTALK (2020). *uTalk - Aprenda cualquier idioma*, consultada el 20 de septiembre de 2020.
Android:
<https://play.google.com/store/apps/details?id=com.eurotalk.uTalk>
iOS:
<https://apps.apple.com/es/app/utalk-aprenda-cualquier-idioma/id1085013031>

XDA DEVELOPERS (2020). *Android Version Distribution statistics will now only be available in Android Studio*. Información de abril de 2020, consultada el 28 de septiembre de 2020.
<*https://www.xda-developers.com/android-version-distribution-statistics-android-studio/*>

XTENSIO (2020). *User Persona Template and Examples*, consultada el 10 de octubre de 2020.
<https://xtensio.com/user-persona/>

YEEPLY (2020). *¿Cuánto cuesta crear una app?* Artículo de junio de 2019, consultado el 2 de octubre de 2020.
<https://www.yeeply.com/blog/cuanto-cuesta-crear-una-app/>

YOUGOV (2015). *Why is blue the world's favorite color?* Consultada el 12 de octubre de 2020.
<https://today.yougov.com/topics/international/articles-reports/2015/05/12/why-blue-worlds-favorite-color>

YUN CHENG & ALDO OLIVARES DOMÍNGUEZ (2019). *Advances Android App Architecture. Real-world app architecture in Kotlin 1.3.* Razeware LLC.

10. Anexos

10.1 Instrucciones de instalación, configuración, compilación y ejecución de la aplicación

Se supone que se va a utilizar la configuración de los servicios de Firebase Authentication, Firebase Realtime Database y Firebase Cloud Storage de este trabajo. Si no fuera así, para crear una nueva configuración, se recomienda seguir las instrucciones descritas en los siguientes apartados de esta memoria:

- 4.1 Inicialización y configuración del proyecto
- 4.2 Transformación del diagrama de clases en una estructura de nodos JSON
- 4.3 Utilización de la aplicación sin conexión a Internet

Llegados a este punto, se da por hecho que se dispone de los servicios de Firebase configurados. El siguiente paso es abrir, sincronizar y reconstruir el proyecto en el entorno de desarrollo integrado Android Studio (para el desarrollo de este proyecto se ha utilizado la versión 4.0.1).

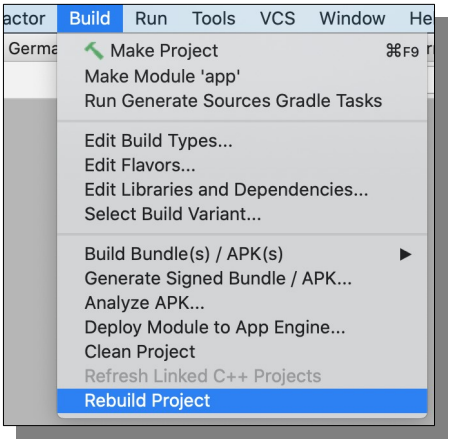

Figura 42: Android Studio Rebuild Project

Indicar que, como se puede observar en la siguiente figura, el archivo build.gradle, además de la configuración del proyecto en la que se define la versión del sdk de compilación, la versión de sdk mínima requerida, etc. contiene la relación de librerías requeridas en forma de dependencias.

```
// *** Firebase ***
// Import the BoM for the Firebase platform
implementation platform('com.google.firebase:firebase-bom:25.12.0')
// Declare the dependency for the Firebase Authentication library
// When using the BoM, you don't specify versions in Firebase library dependencies
implementation 'com.google.firebase:firebase-auth-ktx'
implementation 'com.google.firebase:firebase-database-ktx'
implementation 'com.google.firebase:firebase-storage-ktx'

//RxJava2
implementation 'io.reactivex.rxjava2:rxjava:2.2.20'
implementation 'io.reactivex.rxjava2:rxandroid:2.1.1'

// Multidex
implementation 'androidx.multidex:multidex:2.0.1'

// GlideToVectorYou
implementation 'com.github.corouteam:GlideToVectorYou:v2.0.0'

// PhotoView
implementation 'com.github.chrisbanes:PhotoView:2.0.0'

// Material Drawer
implementation 'com.mikepenz:materialdrawer:7.0.0'
```

Figura 43: Extracto de dependencias del archivo build.gradle

Finalmente, una vez reconstruido el proyecto y finalizada la sincronización de Gradle, se está en disposición de ejecutar la aplicación sobre un dispositivo emulado o real. Para ello, debe seleccionarse la configuración de ejecución y el dispositivo correspondientes y pulsar sobre la opción Run.

Figura 44: Seleccionar configuración y dispositivo, y ejecutar

10.2 Manual de usuario de la aplicación

10.2.1 Introducción

Mediante la Academia virtual de alemán en forma de app el alumnado puede cursar las distintas unidades didácticas que conforman los niveles A1, A2, B1 y B2 de alemán a través de un aprendizaje personalizado, como lo hace una persona nativa y con *feedback* correctivo.

La app resulta muy fácil de utilizar gracias a su interfaz de usuario simple e intuitiva.

En los siguientes apartados se describe el funcionamiento de la aplicación.

10.2.2 Registro

Para registrarse se requiere conexión a Internet y se deben seguir los siguientes pasos:

1. Acceder a la pantalla de registro.
2. Especificar una dirección de correo electrónico.
3. Especificar una contraseña de como mínimo 6 caracteres formada al menos por letras y números.

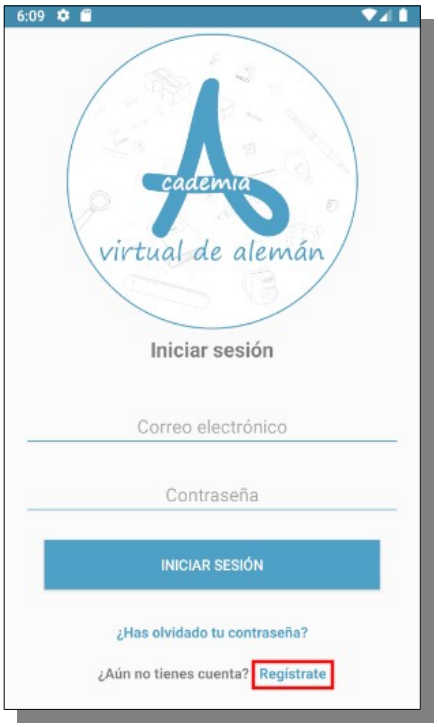

Figura 45: Opción "Regístrate"

Figura 46: Pantalla de registro

10.2.3 Autenticación

Para autenticarse, simplemente debe especificarse el correo electrónico y la contraseña que se han utilizado para registrarse.

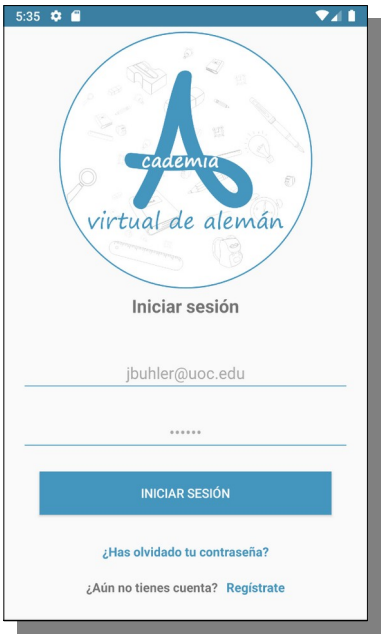

Figura 47: Pantalla de autenticación

10.2.4 Selección de nivel de alemán

Seleccionar el nivel de alemán que se desea cursar.

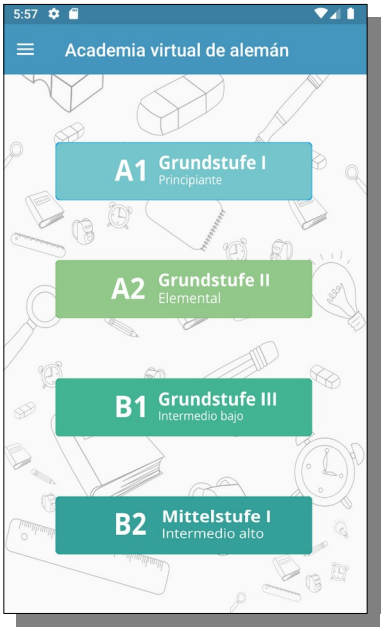

Figura 48: Pantalla de selección de nivel de alemán

10.2.5 Selección de unidad didáctica

Seleccionar la unidad didáctica que se desea cursar.

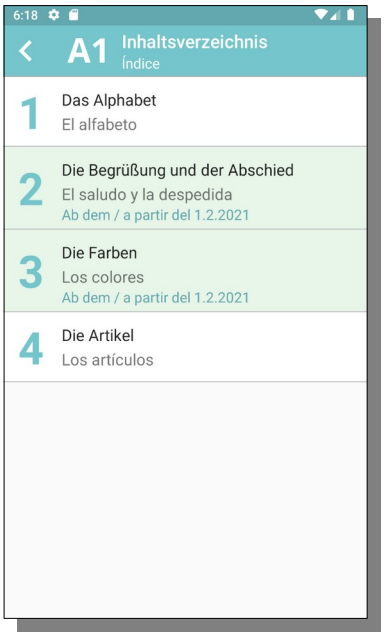

Figura 49: Pantalla de selección de unidad didáctica

10.2.6 Navegación entre páginas y al índice de unidades didácticas

Todas las páginas contienen una barra de navegación en la parte superior, mediante la cuál se puede navegar a la siguiente página, a la página anterior y volver al índice de unidades didácticas.

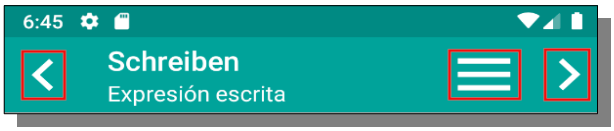

Figura 50: Barra de navegación

10.2.7 Realización de ejercicios

Todas las páginas correspondientes a ejercicios contienen un enunciado, ubicado justo debajo de la barra de navegación, en el que se describe la tarea a realizar.

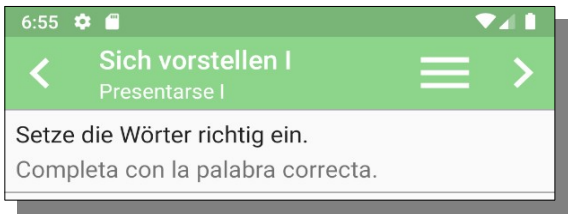

Figura 51: Enunciado del ejercicio

10.2.8 Consulta de créditos y cerrar la sesión

En la pantalla de selección de nivel contiene el menú de la aplicación al cuál se puede acceder mediante el botón hamburguesa, que se encuentra en el extremo superior izquierdo de la pantalla o mediante un desplazamiento con el dedo hacia la derecha iniciado desde el borde izquierdo de la misma.

El menú contiene las opciones para consultar los créditos de la aplicación y para cerrar la sesión.

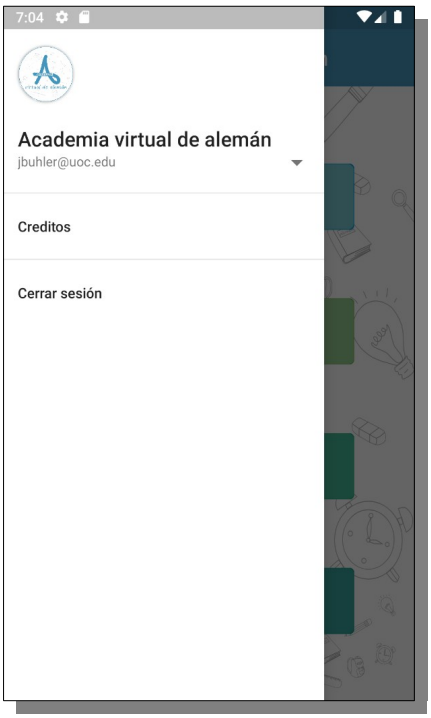

Figura 52: Menú de la aplicación

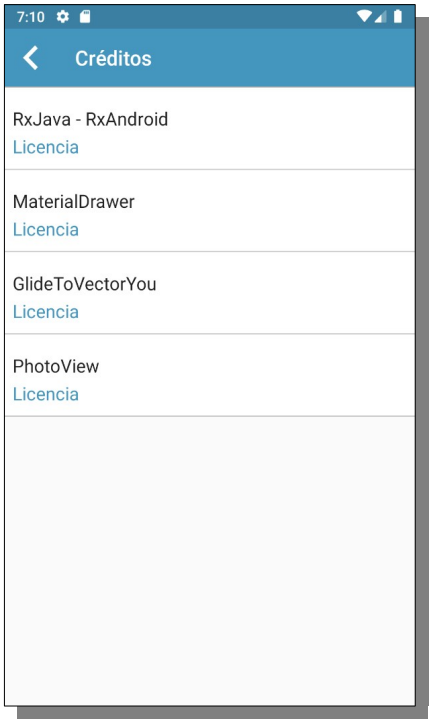

Figura 53: Pantalla de créditos