

Desarrollo de una aplicación móvil como soporte al aprendizaje  
de idiomas  
Mobile application development as a complement to language  
learning

Álvaro Sánchez Torres

GRADO EN INGENIERÍA INFORMÁTICA. FACULTAD DE INFORMÁTICA  
UNIVERSIDAD COMPLUTENSE DE MADRID



Trabajo de fin de grado del Grado en Ingeniería Informática

Madrid, junio de 2022

Director:

Bernabé García, Sergio



# Agradecimientos

Gracias a aquellas personas que me han aportado su grano de arena para desatar mi curiosidad por el desarrollo de aplicaciones móviles.

En especial, gracias a mi director, Sergio, quien se ha preocupado por el estado del proyecto desde el primer día, ayudándome en cada aspecto en el que dudaba y cada problema que me surgía. Gracias por conseguir que este trabajo saliera adelante a tiempo, organizando reuniones cada mes y teniendo el cuidado necesario para que la calidad del proyecto pueda ser la máxima posible. Gracias por aceptar esta propuesta de trabajo final, aportándome una gran libertad a la hora de elegir las funcionalidades del mismo.

Y, por último, gracias a mi familia y a mi novia, que me han estado apoyando en los momentos más difíciles. Quienes han tenido la mayor preocupación por el proyecto, quienes han aportado su granito de arena a la hora de diseñar las funcionalidades finales y quienes querían estar al tanto de cada avance que se conseguía.



# Índice general

Índice general	I
Índice de figuras	IV
Índice de tablas	VI
Resumen	VIII
Abstract	X
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	2
1.3. Plan de trabajo . . . . .	2
1.4. Organización de esta memoria . . . . .	3
<b>2. Estado del arte</b>	<b>6</b>
2.1. Dispositivos móviles . . . . .	6
2.2. Aplicaciones móviles para el aprendizaje de idiomas . . . . .	7
2.2.1. Apps exclusivas de Android . . . . .	8
2.2.2. Apps exclusivas de iOS e iPadOS . . . . .	10
2.2.3. Multiplataforma . . . . .	12
2.2.4. Análisis de los resultados . . . . .	15
2.3. Lenguajes de desarrollo para la aplicación Android . . . . .	16
2.3.1. Java . . . . .	16
2.3.2. Kotlin . . . . .	17

2.3.3. Selección del lenguaje . . . . .	17
<b>3. Diseño de la aplicación</b>	<b>18</b>
3.1. Dispositivo hardware elegido . . . . .	18
3.2. Arquitectura de la aplicación Android . . . . .	19
3.3. Casos de uso . . . . .	20
3.4. Base de datos . . . . .	27
3.5. Seguridad . . . . .	30
<b>4. Implementación</b>	<b>31</b>
4.1. Implementación de funcionalidades: Back-End . . . . .	31
4.1.1. Inicio de sesión y registro . . . . .	31
4.1.2. Introducción de palabras y colecciones . . . . .	31
4.1.3. Prácticas . . . . .	32
4.1.4. Multilenguaje . . . . .	35
4.2. Visualización de las salidas: Front-End . . . . .	36
<b>5. Resultados</b>	<b>43</b>
5.1. Profiling o resultados técnicos . . . . .	43
5.2. Resultados de usuarios . . . . .	46
<b>6. Conclusiones y trabajo futuro</b>	<b>51</b>
6.1. Conclusiones . . . . .	51
6.2. Trabajo futuro . . . . .	52
<b>Bibliografía</b>	<b>56</b>
<b>A. Introduction</b>	<b>57</b>
A.1. Motivation . . . . .	57
A.2. Objectives . . . . .	58

A.3. Work plan . . . . .	58
A.4. Organization of this report . . . . .	59
<b>B. Conclusions and future work</b>	<b>63</b>
B.1. Conclusions . . . . .	63
B.2. Future work . . . . .	64
<b>C. Flujo de uso y Demo de la aplicación</b>	<b>67</b>
C.1. Flujo de uso . . . . .	67
C.1.1. Test . . . . .	67
C.1.2. El intruso . . . . .	68
C.1.3. Desordenado . . . . .	70
C.1.4. Wordle . . . . .	72
C.2. Demo de la aplicación . . . . .	73

# Índice de figuras

1.1. Diagrama de Gantt. . . . .	4
2.1. Captura de pantalla de Aprende idiomas y habla. . . . .	9
2.2. Captura de pantalla de Aprende 100 idiomas. . . . .	10
2.3. Captura de pantalla de LetMeSpeak. . . . .	11
2.4. Captura de pantalla de Go Talk. . . . .	12
2.5. Captura de pantalla de Duolingo. . . . .	13
2.6. Captura de pantalla de Babbel. . . . .	14
2.7. Captura de pantalla de Memrise. . . . .	15
3.1. Modelo de arquitectura de la aplicación. . . . .	19
3.2. <i>Mockup</i> en la herramienta JustInMind. . . . .	20
3.3. Modelo UML de la base de datos. . . . .	28
3.4. Modelo Entidad-Relación de la base de datos. . . . .	29
4.1. Código utilizado para registrar usuario. . . . .	31
4.2. Código utilizado para buscar las palabras pertenecientes a una colección. . . . .	32
4.3. Código utilizado para seleccionar las distintas opciones. . . . .	33
4.4. Código utilizado para marcar respuesta correcta. . . . .	34
4.5. Código utilizado para comprobar una letra en Wordle. . . . .	35
4.6. Vistas de Inicio de sesión y registro. . . . .	36
4.7. Vista de pantalla principal. . . . .	37
4.8. Vista de añadir una palabra. . . . .	38
4.9. Vista de listado de colecciones. . . . .	39
4.10. Vista de listado de palabras. . . . .	39

4.11. Vista de selección de práctica. . . . .	40
4.12. Vista de perfil. . . . .	41
A.1. Gannt Diagram. . . . .	60
C.1. Flujo de uso de la práctica <i>Test</i> . . . . .	68
C.2. Flujo de uso de la práctica <i>Intruso</i> . . . . .	69
C.3. Flujo de uso de la práctica <i>Desordenado</i> . (1) . . . . .	70
C.4. Flujo de uso de la práctica <i>Desordenado</i> . (2) . . . . .	71
C.5. Flujo de uso de la práctica <i>Wordle</i> . . . . .	73

# Índice de tablas

2.1. Comparación de las aplicaciones android disponibles. . . . .	8
2.2. Comparación de las aplicaciones ios disponibles. . . . .	10
2.3. Comparación de las aplicaciones multiplataformas disponibles. . . . .	12
2.4. Comparación de lenguajes para programación en Android. . . . .	16
3.1. Caso de uso 01: Registrar usuario. . . . .	22
3.2. Caso de uso 02: Iniciar sesión. . . . .	22
3.3. Caso de uso 03: Crear colección. . . . .	23
3.4. Caso de uso 04: Registrar palabra. . . . .	23
3.5. Caso de uso 05: Editar palabra. . . . .	24
3.6. Caso de uso 06: Eliminar palabra. . . . .	24
3.7. Caso de uso 07: Visualizar colecciones. . . . .	24
3.8. Caso de uso 08: Eliminar colección. . . . .	25
3.9. Caso de uso 09: Visualizar palabras de una colección. . . . .	25
3.10. Caso de uso 10: Visualizar perfil. . . . .	26
3.11. Caso de uso 11: Jugar a Test, El intruso o Desordenado. . . . .	26
3.12. Caso de uso 12: Jugar a Wordle. . . . .	27
5.1. Medidas de rendimiento en la creación de una palabra. . . . .	44
5.2. Medidas de rendimiento en la edición de una palabra. . . . .	44
5.3. Medidas de rendimiento en la eliminación de una palabra. . . . .	44
5.4. Medidas de rendimiento en la creación de una colección. . . . .	44
5.5. Medidas de rendimiento en la eliminación de una colección. . . . .	45
5.6. Medidas de rendimiento en el listado de las palabras. . . . .	45

5.7. Medidas de rendimiento en el listado de las colecciones. . . . .	45
5.8. Medidas de rendimiento en el registro de un usuario. . . . .	45
5.9. Valoración de Helga. . . . .	47
5.10. Valoración de David. . . . .	48
5.11. Valoración de Javier. . . . .	49
5.12. Valoración de Diego. . . . .	50

# Resumen

Este trabajo de fin de grado tiene como finalidad desarrollar un sistema que de soporte a personas que estén aprendiendo nuevos idiomas, con el objetivo de disponer de un sistema sencillo y elegante en el que anotar las nuevas palabras que el usuario descubra y practicar con ellas más adelante.

Este sistema está compuesto por una aplicación para dispositivos móviles que utilicen sistema operativo Android (teléfonos inteligente o tabletas) que se convertirán en el cuaderno en el que los usuarios almacenen sus palabras y practiquen con ellas.

## Palabras clave

Aprendizaje de idiomas, aplicación móvil, Android, SQLite, Java, XML.

# Abstract

The aim of this final degree project is to develop a system that supports people who are learning new languages, with the objective of having a simple and elegant system in which to write down the new words that the user discovers and practice with them later on.

This system consists of an application for mobile devices that use Android operating system (smartphones or tablets) that will become the notebook in which users store their words and practice with them.

## Keywords

Language learning, mobile application, Android, SQLite, Java, XML.



# Capítulo 1

## Introducción

### 1.1. Motivación

Mantener la concentración en una clase suele ser complicado, más aún cuando se trata de introducirse en un idioma nuevo para el alumno. Muchas veces es frustrante la nula comprensión del vocabulario y se acaba dejando de prestar atención al profesor e incluso abandonando el objetivo propuesto.

Siempre se ha recurrido a la utilización de juegos en el ámbito educativo para mejorar la concentración del alumnado [1], hasta el punto de que, con el avance de las tecnologías, muchas personas han dejado de asistir a academias (o como en este caso concreto, a escuelas de idiomas) pensando que con una aplicación móvil o una página web podrán conseguir lo mismo [2].

Sin embargo, la presencia de un profesor sigue siendo muy importante para el aprendizaje, puesto que puede comprender cómo se siente el alumno, reconocer sus fortalezas y sus debilidades y poder actuar en consecuencia para eliminar sus carencias.

En este proyecto se valora la cooperación del profesorado con las nuevas tecnologías, desarrollando una herramienta que les permita distribuir el vocabulario más importante para cada persona e incentivar a sus alumnos a tomar la iniciativa a la hora de seguir desarrollando su aprendizaje por su propia cuenta [3].

Además, también está pensada para las personas que quieran mantener un ambiente

autodidacta. Estas podrían organizar las palabras que descubran en el día a día: leyendo libros, viendo películas o escuchando canciones, entre otras actividades. Esta aplicación permite anotar estas palabras y practicar con ellas en cualquier lugar. Por este motivo se determinó que LiMob podría ser un gran nombre, ya que “Li” deriva de Linguae y de Libello (“Idiomas” y “Cuaderno” en latín) y “Mob” deriva de Mobile (“Móvil” en inglés).

## 1.2. Objetivos

El objetivo principal del proyecto es desarrollar una aplicación para dispositivos móviles con sistema operativo Android. Esta aplicación permitirá a sus usuarios organizar traducciones de palabras y practicar con ellas mediante juegos. Para ello, se desarrollarán unas funcionalidades específicas previamente definidas. Estos objetivos incluyen:

- Desarrollo de una aplicación para dispositivos móviles con sistema operativo Android.
- Desarrollo e implementación de una base de datos local para cada usuario donde almacenar toda la información relacionada con sus palabras y algunos detalles de sus últimas prácticas.
- Desarrollo e implementación mediante Java y XML de juegos interactivos donde poder practicar las palabras almacenadas en la base de datos.
- Uso del entorno de desarrollo integrado Android Studio, en el que se hará uso del lenguaje de programación Java para realizar un diseño orientado a objetos adecuado a nuestras necesidades.

## 1.3. Plan de trabajo

Para comenzar se realizó un trabajo de investigación sobre las plataformas existentes dedicadas a dar soporte al aprendizaje de idiomas. Para ello, se descargaron la mayoría de aplicaciones destacables y se hicieron pruebas de uso, observando las funcionalidades

que ofrecían al usuario. También se realizó un trabajo de investigación sobre las distintas opciones de desarrollo de aplicaciones Android.

A continuación se quiso dejar claras las funcionalidades que debería de tener la aplicación y los datos que se deberían utilizar para que la utilización de la aplicación fuese lo más amena posible. Además, se realizó un estudio sobre los dos lenguajes utilizados (Java y XML) que sirvió de ayuda a la hora de comenzar el desarrollo.

Para poder avanzar de forma consistente en el proyecto, se diseñó la base de datos y se realizó un mockup en el que se asentaban las bases que tendría la interfaz del proyecto final.

Lo que acaparó la mayor parte del tiempo fue la implementación de la aplicación. Añadiendo algunos detalles que no se tuvieron en cuenta en las fases anteriores, pero que a medida que iba avanzando el desarrollo, se observó que podrían ser de gran ayuda para el usuario. El código se encuentra disponible en el siguiente repositorio de GitHub<sup>1</sup>.

Finalmente, se concluyó con el desarrollo de esta memoria, con el fin de plasmar todo el desarrollo del proyecto y dejar constancia de las decisiones tomadas a lo largo del trabajo.

Observando la Figura 1.1, podemos entender, gracias a un diagrama de Gannt, la cantidad de tiempo dedicada a cada uno de los bloques del desarrollo. Se ha utilizado el color salmón para destacar las actividades relacionadas con la investigación de las herramientas para el futuro desarrollo del proyecto, el azul para las relacionadas con la especificación del proyecto, el amarillo para las relacionadas con el diseño, el rosa para las relacionadas con la implementación y el gris para la memorial final.

## 1.4. Organización de esta memoria

En este apartado se describe el contenido y la organización de los apartados del documento:

- **Capítulo 2:** Contiene un estudio de mercado de las aplicaciones más relevantes que se pueden descargar en este momento, que presentan funcionalidades y naturalezas

---

<sup>1</sup><https://github.com/AlvaroSan98/TFGIIdiomas>

# Plan de trabajo

2021/2022

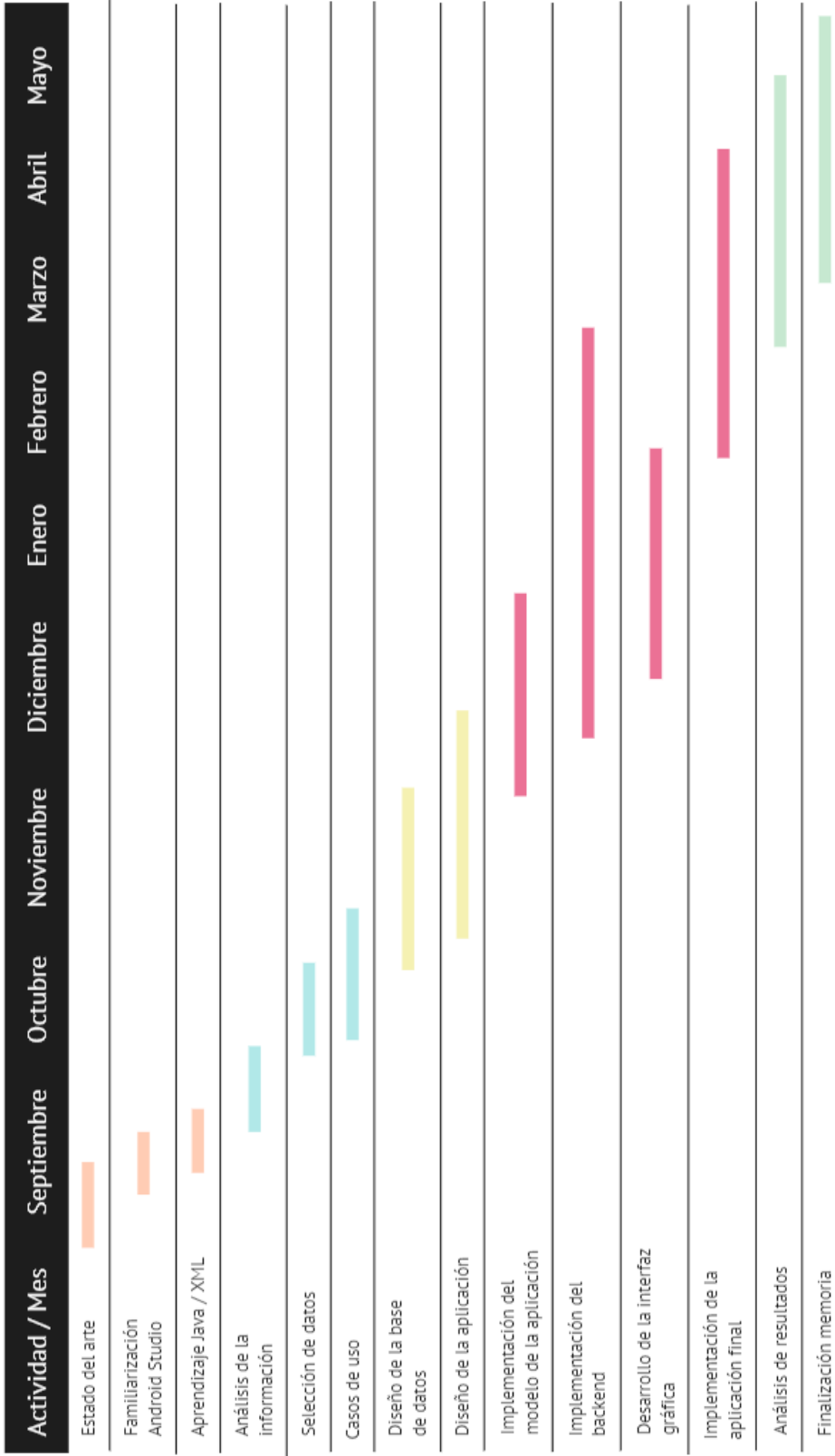


Figura 1.1: Diagrama de Gantt.

parecidas a LiMob además de un análisis de los resultados observados tras dicho estudio. A continuación se manifiesta una breve descripción de las tecnologías empleadas en el desarrollo de LiMob, junto a razones de elección de las mismas.

- **Capítulo 3:** Se presentan algunos de los casos de uso más relevantes que los usuarios pueden llevar a cabo en la aplicación, seguido de el modelo de la base de datos empleado y la arquitectura que presenta la aplicación.
- **Capítulo 4:** Se detalla todo el desarrollo de la aplicación, el diseño de la interfaz de usuario y los componentes añadidos a sus diferentes pantallas.
- **Capítulo 5:** Trata de un análisis de las métricas obtenidas del uso de la aplicación y comentarios recibidos por usuarios que han sido expuestos a la misma.
- **Capítulo 6:** Expone las conclusiones obtenidas durante el desarrollo de este proyecto y los posibles trabajos futuros de mejora, incremento o iteración de funcionalidades.
- En último lugar se encuentra la bibliografía y los anexos donde se incluye una guía de uso de la aplicación y un vídeo demostrativo.

# Capítulo 2

## Estado del arte

Después de definir la motivación y el propósito del proyecto, se van a analizar las diferentes plataformas para el desarrollo de esta aplicación y el mercado de aplicaciones de cada una de ellas. El objetivo de este capítulo es ofrecer una breve descripción de las aplicaciones ya disponibles, distinguiendo sus virtudes y carencias, y analizar las plataformas y los sistemas existentes que se pueden utilizar para el desarrollo de esta aplicación.

### 2.1. Dispositivos móviles

Actualmente los dispositivos móviles están muy presentes en la vida de las personas, ya que estos pueden ofrecer características muy similares a las de los grandes ordenadores sin necesitar tanto espacio. Entre ellos podemos distinguir entre simples teléfonos, dispositivos un poco más grandes, como el caso de las tablets, u otros mucho más pequeños, como relojes o pulseras que miden nuestra calidad de vida. A continuación se describirán los detalles de cada tipo de dispositivo:

- **Tablets:** Son dispositivos de gran tamaño (entre 7 y 13 pulgadas aproximadamente) que pueden disponer o no de red de telefonía para llamadas, mensajes y acceso a internet sin la necesidad de estar conectado a una red WiFi. El primer objetivo de estos dispositivos fue el entretenimiento, pero dada su facilidad de uso y la manera tan sencilla de transportarlos, actualmente es común verlos incorporados al mundo

laboral. Algunos de estos dispositivos ofrecen un lapicero muy preciso, con el que se convierten en la mejor herramienta para tareas complicadas como el diseño gráfico o algunas otras más sencillas como tomar apuntes.

- **Smartphones:** Estos dispositivos tienen un tamaño medio (entre 4 y 7 pulgadas aproximadamente) ya que su principal objetivo es poder transportarlos con nosotros a cualquier lugar. Combinan las funcionalidades de un teléfono móvil (ya que son la evolución directa de ellos) y otras que se encontraban en ordenadores de bolsillo o PDAs. Además, ofrecen otras funcionalidades propias, como las videollamadas, no disponibles en ninguno de sus predecesores.
- **Smartwatches:** Estos dispositivos son similares a los clásicos relojes en cuanto a diseño. Su principal característica es que disponen de una pantalla táctil, comunmente a color, en la que poder manejar sus distintas aplicaciones destinadas a la salud y al deporte en la mayoría de los casos. Además pueden estar vinculados a un smartphone, de esta forma se puede visualizar las notificaciones recibidas directamente en su pequeña pantalla o realizar pagos en lectores contactless.
- **Wearables:** Estos dispositivos se incorporan en alguna parte del cuerpo interactuando de forma continua con el usuario y con otros dispositivos con la finalidad de realizar alguna función concreta. Suelen estar destinados a la salud y el deporte, y aunque no se suele poder interactuar con ellos de forma directa, cuentan con una obtención precisa de datos que envían a un smartphone para poder procesarlos y visualizarlos.

## 2.2. Aplicaciones móviles para el aprendizaje de idiomas

En este apartado se van a analizar las aplicaciones disponibles actualmente en el mercado destinadas al aprendizaje de idiomas, ya sea de forma interactiva o tradicional.

### 2.2.1. Apps exclusivas de Android

La búsqueda de aplicaciones disponibles para Android se ha realizado en la tienda de aplicaciones de esta plataforma, Play Google. Se han encontrado distintas opciones interesantes las cuales se analizan a continuación. Adicionalmente, en la Tabla 2.1 se comparan las diferentes aplicaciones analizadas.

	Aprende idiomas y habla	Aprende 100 idiomas
Vocabulario preestablecido	Sí	Sí
Prueba de nivel	No	No
Mecánicas de juego	Sí	Sí
Gratuita	Sí	Sí
Anuncios	Sí	Sí
Descargas	50.000+	100.000+
Última actualización	07/02/2022	31/03/2021

Tabla 2.1: Comparación de las aplicaciones android disponibles.

- **Aprende idiomas y habla:** Es una aplicación destinada a aprender idiomas para personas de habla inglesa. Tiene mecánicas de juego que hacen uso de comparaciones entre palabras en inglés y su respectiva traducción en el idiomas que hayamos escogido. A día 12 de mayo de 2022 existen 6 idiomas para practicar.

Esta aplicación hace bastante énfasis en la práctica del idioma hablado. Además de los juegos en los que tendremos que seleccionar la traducción adecuada para una palabra o una imagen descriptiva, tiene diferentes juegos que hacen uso del micrófono. Por ejemplo, uno de ellos es una profesora virtual que traducirá cualquier frase que se le diga.

Uno de los juegos es un test en el cual debemos seleccionar la traducción correcta a una palabra entre 4 opciones disponibles. Sin embargo, donde debería mostrarse la palabra a traducir, no se puede ver nada, tal y como podemos observar en la Figura 2.1.

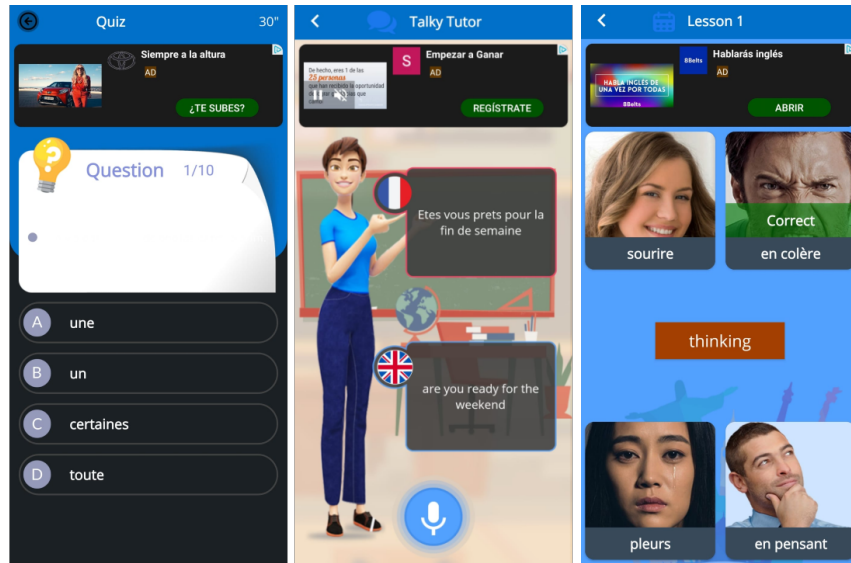


Figura 2.1: Captura de pantalla de Aprende idiomas y habla.

- Aprende 100 idiomas:** Es una aplicación destinada a aprender el vocabulario de idiomas concretos. A día 22 de mayo de 2022 cuenta con 100 idiomas que podemos practicar. Se enfoca en aportar una gran cantidad de palabras las cuales poder aprender viendo su traducción directa, una imagen descriptiva o un audio dictándola.

Además, no solo aporta palabras, si no que también aporta frases. De esta forma, el aprendizaje se hará más útil para personas que quieran lanzarse a hablar el idioma nada más comiencen a aprenderlo.

Aparte, utilizan diferentes juegos para poder practicar las palabras descubiertas. Ejemplos de los mismos pueden ser tests, elegir la imagen correcta después de escuchar un audio, o elegir la palabra que falta en una frase (como podemos observar en la Figura 2.2).

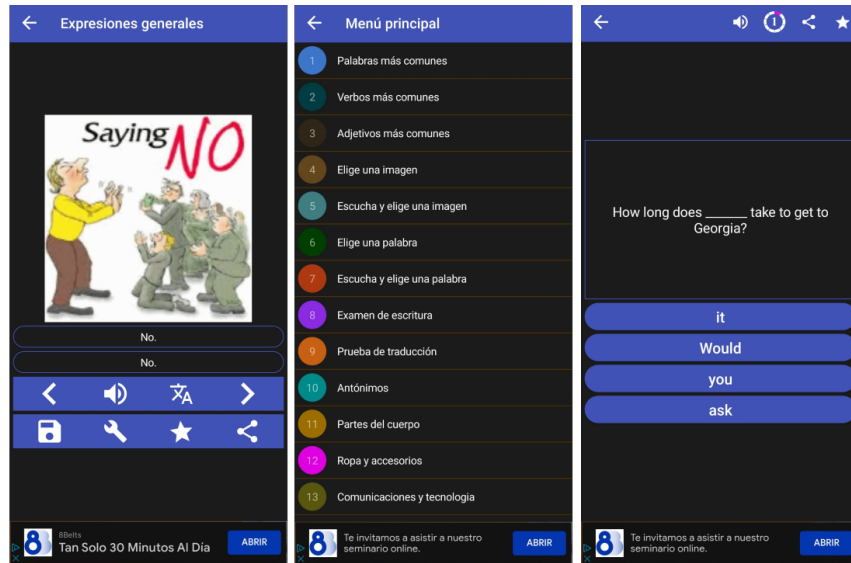


Figura 2.2: Captura de pantalla de Aprende 100 idiomas.

### 2.2.2. Apps exclusivas de iOS e iPadOS

La búsqueda de aplicaciones disponibles para iOS se ha realizado en la tienda de aplicaciones de esta plataforma, App Store. Se han encontrado algunas aplicaciones interesantes, las cuales se analizan a continuación. En la Tabla 2.2 se comparan estas aplicaciones.

	LetMeSpeak	Go Talk
Vocabulario preestablecido	Sí	Sí
Prueba de nivel	Sí	No
Mecánicas de juego	Sí	Sí
Gratuita	Sí	Limitada
Anuncios	No	No

Tabla 2.2: Comparación de las aplicaciones ios disponibles.

- LetMeSpeak:** Se trata de una aplicación destinada al aprendizaje autodidacta de un idioma. Nada más instalarla, hace una prueba de nivel al usuario y permite seleccionar las categorías que más le interesa aprender (hacer un viaje o coger el transporte público, entre otras).

Además, la aplicación cuenta con una metodología de juegos bastante integrada. Cada

práctica que el usuario realice le otorgará estrellas, las cuáles podrá canjear en un futuro por nuevas categorías para seguir avanzando con su aprendizaje.

Además, tal y como podemos observar en la Figura 2.3, el usuario se encuentra siempre en punto de un mapa establecido por la aplicación, y según su progreso con las prácticas que realice podrá avanzar su posición o quedarse en el mismo sitio.

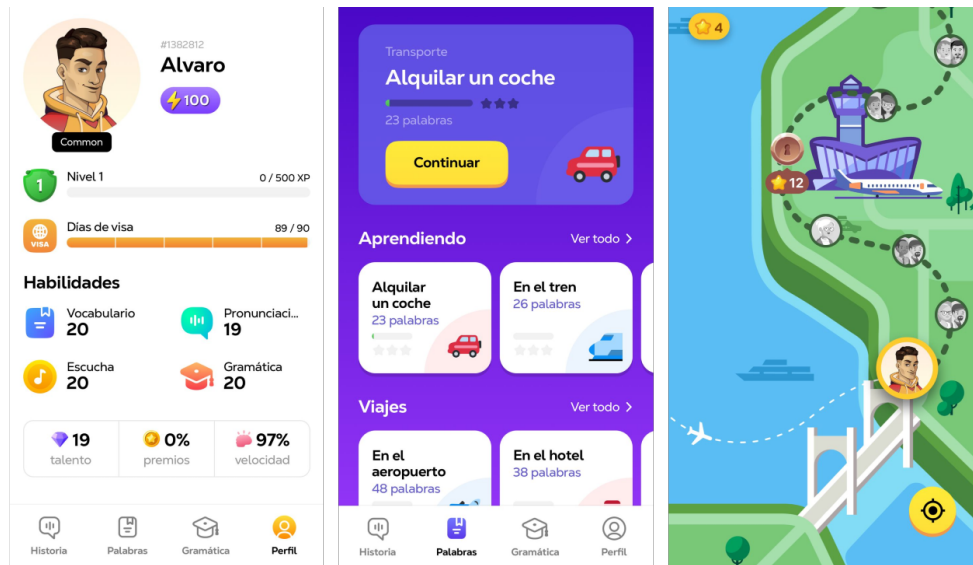


Figura 2.3: Captura de pantalla de LetMeSpeak.

- **Go talk:** Esta aplicación cuenta con un gran catálogo de idiomas que el usuario puede aprender. Aunque todos se pueden comenzar a estudiar gratuitamente, solamente cuenta con 1 tema gratuito, y una vez que el usuario lo complete debería de comprar más temas si quiere seguir avanzando.

Utiliza una metodología de juegos para enseñar el vocabulario, y se centra fundamentalmente en el uso de los altavoces y el micrófono a la hora de realizar las prácticas. Estos juegos, como podemos observar en la Figura 2.4, pueden ser: mostrar una foto, su correspondiente palabra en inglés, y que el sistema reconozca la palabra que el usuario ha dicho a través del micrófono; o que el usuario tenga que seleccionar la palabra que la aplicación ha reproducido por los altavoces, entre otros.



Figura 2.4: Captura de pantalla de Go Talk.

### 2.2.3. Multiplataforma

En este apartado se incluyen las aplicaciones multiplataformas que soportan tanto Android, como iOS o plataformas webs. Tal y como se ha realizado para las aplicaciones destinadas exclusivamente a Android o iOS, en la Tabla 2.3 se ha hecho una comparación visual de las aplicaciones analizadas.

	Duolingo	Babbel	Memrise
<b>Vocabulario preestablecido</b>	Sí	Sí	Sí
<b>Prueba de nivel</b>	Sí	Sí	No
<b>Mecánicas de juego</b>	No	Sí	No
<b>Tipo</b>	Gratis	Pago	Gratis
<b>Gratis limitado</b>	No	-	Sí
<b>Descargas</b>	100.000.000+	10.000.000+	10.000.000+
<b>Última actualización</b>	09/05/2022	10/05/2022	07/04/2022

Tabla 2.3: Comparación de las aplicaciones multiplataformas disponibles.

- **Duolingo:** Es la aplicación destinada a aprender idiomas más descargada en la actualidad [4]. Esta plataforma utiliza juegos prediseñados para practicar el aprendizaje de distintos idiomas. Está disponible tanto en versión web como en versión móvil

(teniendo soporte en Android, iOS y Windows Phone).

Una vez registrado se selecciona el idioma que queremos aprender, pudiendo elegir también la utilización de diferentes alfabetos (chino o árabe, entre otros) como podemos observar en la Figura 2.5. Antes de poder practicar con la aplicación, se hace un examen con el que poder valorar nuestro nivel lingüístico del idioma, con la finalidad de poder empezar a practicar con juegos más complejos según nuestro dominio de esa lengua.

Una vez dentro del curso, hay disponibles distintas actividades (traducir frases emparejando palabras, reconocimiento oral de frases reproducidas en la aplicación, etc) agrupadas en diferentes categorías.

Nuestro progreso en el curso se identifica mediante un gráfico que determina nuestro grado de dominio de las distintas unidades. Además, Duolingo nos ofrece un sistema de logros con los que fomentar nuestro interés en dedicar más tiempo al aprendizaje.



Figura 2.5: Captura de pantalla de Duolingo.

- **Babbel**: es una aplicación completamente destinada al aprendizaje autodidáctica de idiomas. Dispone de distintas lecciones y cursos con los que desarrollar nuestro vocabu-

lario. Además, incorpora imágenes a los ejercicios con la finalidad de mejorar nuestra comprensión y tener una mayor retención del conocimiento adquirido.

Aparte de los ejercicios interactivos, existe un catálogo de podcasts que servirán de ayuda para progresar en las lecciones y otros dos juegos en los que se utiliza el vocabulario aprendido para sortear obstáculos con un personaje. Estas funcionalidades se pueden observar en la Figura 2.6.

A día 19 de abril de 2022 Babel dispone de 14 lenguas para aprender y está disponible tanto en versión web como en versión móvil (Android e iOS).

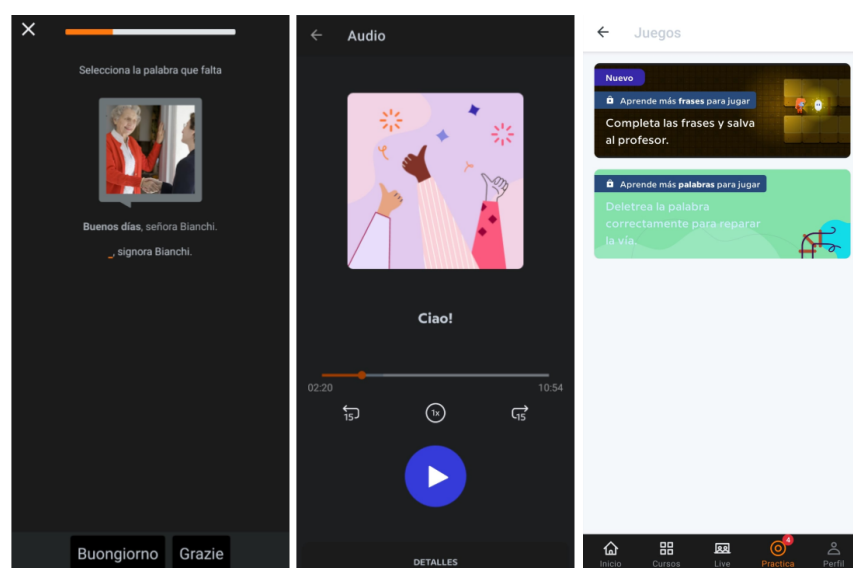


Figura 2.6: Captura de pantalla de Babel.

- **Memrise:** es una aplicación disponible tanto en versión web como en versión móvil (iOS y Android) destinada al aprendizaje de idiomas mediante el uso de la psicología. Utiliza flash cards (mostradas en la Figura 2.7 aumentadas con reglas mnemotécnicas para facilitar en mayor medida el aprendizaje del lenguaje. Permite elegir al usuario la regla mnemotécnica con la que recordar las palabras preestablecidas que dispone la aplicación.

Además, desde Memrise se puede especificar la finalidad del lenguaje que queremos

adquirir, permitiéndonos estudiar el vocabulario específico que requerirá un turista u otro más cercano con el que conversar con una familia nativa.

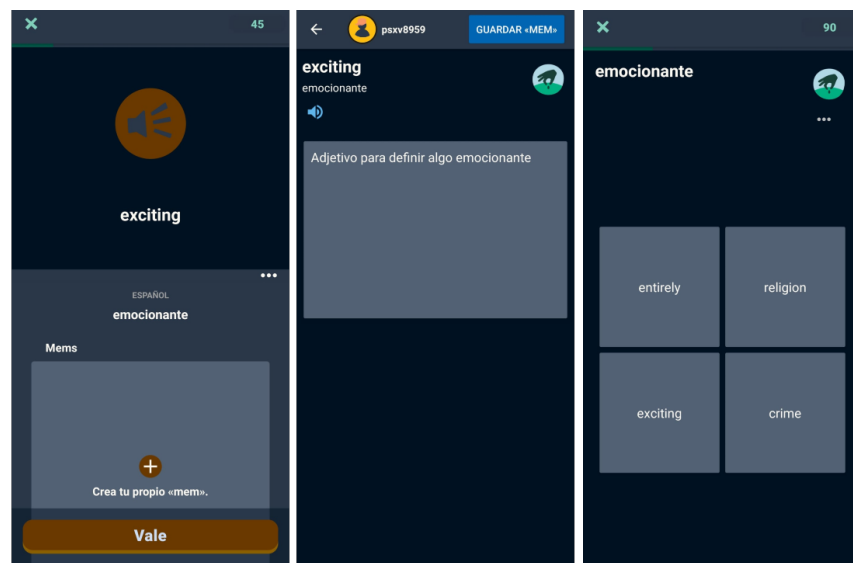


Figura 2.7: Captura de pantalla de Memrise.

#### 2.2.4. Análisis de los resultados

El aprendizaje de idiomas es un gran nicho de mercado, por lo que bastantes empresas con grandes presupuestos desarrollan aplicaciones multiplataforma con las que captar clientes. Como hemos podido observar comparando las descargas, éstas aplicaciones dejan completamente de lado a las soportadas únicamente en un solo sistema operativo. Aunque si nos enfocamos en cada uno de los dos mercados, existe un mayor número de descargas en Google Play [5].

Todas ellas están enfocadas en un aprendizaje ya establecido, sin dotar al usuario de mucha personalización e incluso volviéndose un poco repetitivas si alguna de las lecciones ya es conocida por el usuario. Sin embargo, LiMob dota al usuario de una completa personalización de los ejercicios, ya que solamente las palabras almacenadas en las colecciones agregadas son las utilizadas en las prácticas. Además, si el usuario cree que una colección la domina al completo puede eliminarla.

Las aplicaciones mencionadas cuentan con numerosos ejercicios con los que retener la atención del usuario. Sin embargo, a la larga resultan repetitivos y monótonos, ya que (exceptuando los juegos interactivos de Babel) no tienen un sistema que permita al usuario superarse cada vez que inicia un juego nuevo. En ese aspecto, LiMob cuenta con un sistema de vidas o tiempo que permite al usuario implicarse a fondo a la hora de practicar.

## 2.3. Lenguajes de desarrollo para la aplicación Android

Dada la dificultad en desarrollar una aplicación móvil desde 0 siendo un solo integrante en el grupo y sin haber practicado anteriormente, se ha optado por desarrollar la aplicación exclusivamente para Android. Debido a que la gran mayoría de aplicaciones para smartphones están desarrolladas para este sistema operativo, se ha considerado que existe una mayor documentación que facilitaría el desarrollo de la aplicación. Además, aunque en la asignatura optativa Programación de aplicaciones para dispositivos móviles se enseña a desarrollar tanto aplicaciones Android como aplicaciones multiplataforma, se enfoca mucho más en la primera opción. Por tanto, barajamos los 2 lenguajes de desarrollo en Android más conocidos [6]. Podemos observar una pequeña comparación preliminar en la Tabla

2.4.

	<b>Java</b>	<b>Kotlin</b>
<b>Diseñador</b>	Sun Microsystems (Oracle Corporation)	JetBrains
<b>Fundado</b>	1996	2016
<b>Paradigma</b>	Orientado a objetos	Orientado a objetos
<b>Conocimientos previos</b>	Sí	No

**Tabla 2.4:** *Comparación de lenguajes para programación en Android.*

### 2.3.1. Java

Java es uno de los lenguajes de programación más utilizados en el mundo. Fue fundado en 1996, por lo que, tanto su antigüedad como su abundante uso, hace que su comunidad sea uno de los mejores apoyos a la hora de programar [7]. Además, su documentación es

extensa y como ventaja adicional está el conocimiento de este lenguaje de programación al ser usado en la asignatura de Tecnología de la Programación de la carrera.

### **2.3.2. Kotlin**

Kotlin es un lenguaje de programación de tipo estático que corre sobre la máquina virtual de Java y que también puede ser compilado a código fuente de JavaScript. Este lenguaje surgió con la idea de ofrecer una compilación tan rápida como Java con unas características similares a las de Scala (un lenguaje de programación multi-paradigma diseñado para expresar patrones comunes de programación de forma concisa, elegante y con tipos seguros). En 2017 fue considerado por Google como lenguaje oficial para las aplicaciones Android, al mismo nivel que Java.

### **2.3.3. Selección del lenguaje**

Debido al alto grado de conocimiento que se obtuvo en la asignatura de Tecnología de la Programación, el apoyo que generó la asignatura Programación de aplicaciones para dispositivos móviles [8–10] y la diferencia entre las dos comunidades, se ha optado por utilizar Java en el desarrollo de la aplicación. La extensa documentación y el apoyo de sus usuarios podrían ser de gran ayuda a la hora de elaborar el proyecto.

# Capítulo 3

## Diseño de la aplicación

En este capítulo se va a hablar en detalle de las plataformas de destino finales, los diseños de arquitectura de la aplicación, los casos de uso de esta, y el diseño final de su base de datos local.

### 3.1. Dispositivo hardware elegido

Con el fin de facilitar el registro de nuevas palabras en cualquier sitio, se ha decidido hacer una aplicación móvil para smartphones que soporta el sistema operativo más utilizado en la actualidad, Android. Esto se ha decidido así ya que el diseño de este tipo de dispositivos ofrece una serie de características interesantes como: una pantalla táctil, que ofrece una forma de navegación sencilla; es un dispositivo portable, por tanto se puede acceder a la aplicación en cualquier lugar; y, por último, conexión a Internet, lo que nos permite interactuar con otros usuarios en los juegos multijugador.

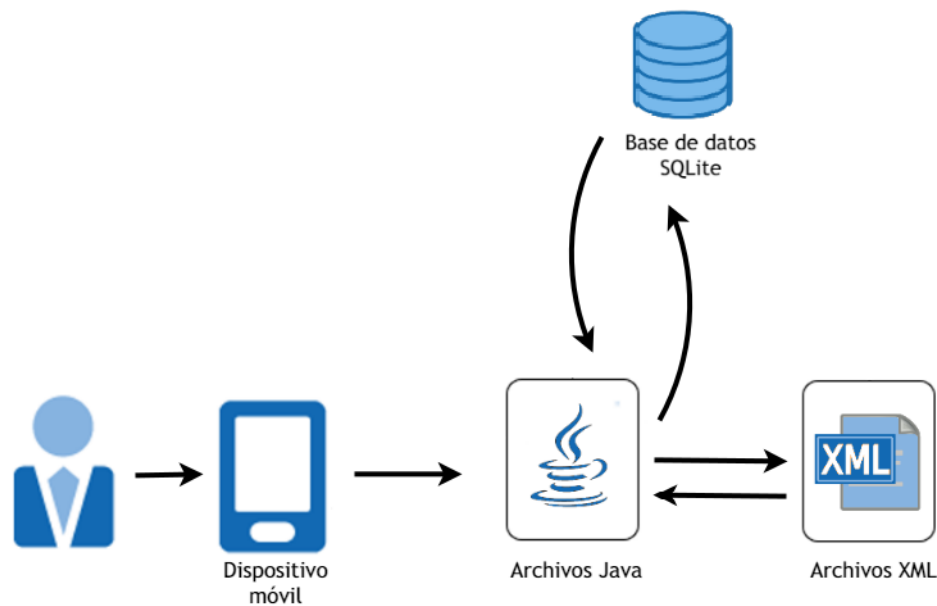
Dado que la edad del público objetivo de la aplicación es demasiado amplia (pueden querer aprender idiomas tanto niños, como adolescentes o adultos) se ha optado por utilizar la API 21 (Lollipop) como el SDK mínimo para poder soportar la aplicación. Según los datos oficiales de Google, el 99 % de los dispositivos Android son capaces de ejecutar aplicaciones con esta API, de esta forma no limitamos el uso de la aplicación a personas con un dispositivo demasiado antiguo o no actualizado [11].

## 3.2. Arquitectura de la aplicación Android

En esta sección se va a explicar las tecnologías utilizadas dentro de la aplicación. Como se puede observar en la Figura 3.1, la parte del modelo correspondería con el código Java utilizado, el cual se comunica con el sistema de base de datos SQLite para proveer al usuario de la información requerida.

Las vistas corresponderían a parte de los archivos XML utilizados. Estos están en continua comunicación con la programación dinámica de la aplicación, ya que dependiendo de los parámetros de la actividad ejecutada las vistas podrían verse modificadas.

Además, la aplicación también hace uso del patrón de programación Observable [12]. Este patrón facilita mucho el chequeo del progreso del usuario en ciertas practicas.



**Figura 3.1:** *Modelo de arquitectura de la aplicación.*

Adicionalmente, se desarrolló un *mockup* del diseño de la aplicación mediante la herramienta JustInMind. Algunas pantallas de este *mockup* se pueden observar en la Figura 3.2.

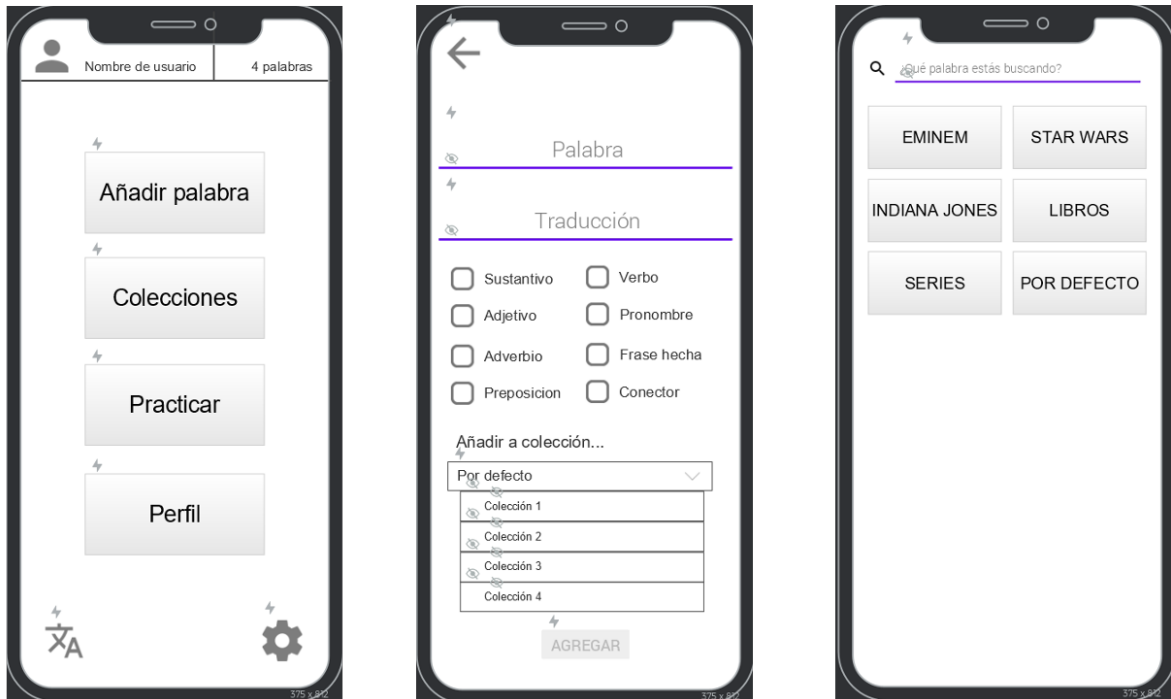


Figura 3.2: Mockup en la herramienta JustInMind.

### 3.3. Casos de uso

En esta aplicación se puede observar la existencia de un solo actor, el usuario que desea aprender idiomas. Para este actor disponemos de los siguientes casos de uso:

- Caso de uso de registrar usuario, reflejado en la Tabla 3.1, muestra los pasos a realizar para crear un usuario dentro de la aplicación.
- Caso de uso de iniciar sesión, reflejado en la Tabla 3.2, muestra los pasos necesarios para iniciar sesión en la aplicación.
- Caso de uso de crear una colección, reflejado en la Tabla 3.3, muestra los pasos necesarios para crear una colección en la base de datos.
- Caso de uso de añadir una palabra, reflejado en la Tabla 3.4, muestra los pasos necesarios para añadir una palabra en la base de datos.

- Caso de uso de editar una palabra, reflejado en la Tabla 3.5, muestra los pasos necesarios para editar una palabra ya existente.
- Caso de uso de eliminar una palabra, reflejado en la Tabla 3.6, muestra los pasos necesarios para eliminar una palabra ya existente.
- Caso de uso de visualizar las colecciones, reflejado en la Tabla 3.7, muestra los pasos necesarios para visualizar las colecciones existentes en la base de datos.
- Caso de uso de eliminar una colección, reflejado en la Tabla 3.8, muestra los pasos necesarios para eliminar una colección ya existente en la base de datos.
- Caso de uso de visualizar las palabras pertenecientes a una colección, reflejado en la Tabla 3.9, muestra los pasos necesarios para visualizar las palabras pertenecientes a una colección específica.
- Caso de uso de visualizar perfil, reflejado en la Tabla 3.10, muestra los pasos necesarios para visualizar el perfil del usuario.
- Caso de uso de jugar a Test, El intruso o Desordenado, reflejado en la Tabla 3.11, muestra los pasos necesarios para jugar uno de los juegos mencionados anteriormente.
- Caso de uso de jugar a Wordle, reflejado en la Tabla 3.12, muestra los pasos necesarios para jugar "Wordle".

<b>CU-01</b>	<b>Registrar usuario</b>
<b>Descripción</b>	Registrar usuario en la base de datos
<b>Autor</b>	Usuario
<b>Entradas</b>	Nombre de usuario Contraseña Idioma a estudiar
<b>Salidas</b>	Redirección a la pantalla principal
<b>Precondición</b>	Tener todos los campos rellenos No debe existir un usuario con el mismo nombre de usuario
<b>Secuencia</b>	Validar los datos introducidos En caso de error mostrar este Comprobar que el nombre de usuario no existe Registrar usuario en la base de datos Redireccionar a la pantalla principal de la aplicación
<b>Postcondición</b>	Usuario registrado

**Tabla 3.1:** *Caso de uso 01: Registrar usuario.*

<b>CU-02</b>	<b>Iniciar sesión</b>
<b>Descripción</b>	Iniciar sesión en la aplicación
<b>Autor</b>	Usuario
<b>Entradas</b>	Nombre de usuario Contraseña
<b>Salidas</b>	Redirección a la pantalla principal
<b>Precondición</b>	Tener todos los campos rellenos El nombre de usuario debe existir en la base de datos La contraseña debe corresponder al nombre de usuario
<b>Secuencia</b>	Validar los datos introducidos En caso de error mostrar este Comprobar que el nombre de usuario existe Comprobar que la contraseña pertenece a ese nombre de usuario Redireccionar a la pantalla de selección de idioma
<b>Postcondición</b>	Token de usuario de la aplicación

**Tabla 3.2:** *Caso de uso 02: Iniciar sesión.*

<b>CU-03</b>	<b>Crear una colección</b>
<b>Descripción</b>	Crear una colección en la base de datos
<b>Autor</b>	Usuario
<b>Entradas</b>	Nombre de la colección
<b>Salidas</b>	Redirección a la pantalla de creación de palabra
<b>Precondición</b>	Tener el campo relleno No debe existir una colección con el mismo nombre en ese idioma
<b>Secuencia</b>	Validar los datos introducidos En caso de error mostrar este Comprobar que el nombre de la colección no existe Registrar colección en la base de datos Redireccionar a la pantalla de creación de palabra
<b>Postcondición</b>	Colección creada

**Tabla 3.3:** *Caso de uso 03: Crear colección.*

<b>CU-04</b>	<b>Añadir palabra</b>
<b>Descripción</b>	Añadir palabra a una colección
<b>Autor</b>	Usuario
<b>Entradas</b>	Palabra Traducción Tipo de palabra Colección a la que añadir
<b>Salidas</b>	Misma pantalla con todos los campos vacíos
<b>Precondición</b>	Tener todos los campos rellenos No debe existir en esa colección la misma palabra y traducción
<b>Secuencia</b>	Validar los datos introducidos En caso de error mostrar este Comprobar que la palabra y su traducción no existen en esa colección Registrar palabra en la base de datos Vaciar todos los campos
<b>Postcondición</b>	Palabra registrada

**Tabla 3.4:** *Caso de uso 04: Registrar palabra.*

<b>CU-05</b>	<b>Editar palabra</b>
<b>Descripción</b>	Editar palabra de una colección
<b>Autor</b>	Usuario
<b>Entradas</b>	Palabra Traducción Tipo de palabra
<b>Salidas</b>	Redirección a la pantalla de la colección
<b>Precondición</b>	Tener todos los campos rellenos
<b>Secuencia</b>	Validar los datos introducidos En caso de error mostrar este Actualizar palabra en la base de datos Redireccionar a la pantalla de la colección
<b>Postcondición</b>	Palabra actualizada

**Tabla 3.5:** *Caso de uso 05: Editar palabra.*

<b>CU-06</b>	<b>Eliminar palabra</b>
<b>Descripción</b>	Eliminar palabra de una colección
<b>Autor</b>	Usuario
<b>Entradas</b>	Palabra a eliminar
<b>Salidas</b>	Eliminar la palabra de la pantalla de colección
<b>Precondición</b>	-
<b>Secuencia</b>	Seleccionar palabra a eliminar Mostrar diálogo de confirmación Eliminar palabra de la base de datos Actualizar la pantalla de la colección sin la palabra eliminada
<b>Postcondición</b>	Palabra eliminada

**Tabla 3.6:** *Caso de uso 06: Eliminar palabra.*

<b>CU-07</b>	<b>Visualizar colecciones</b>
<b>Descripción</b>	Visualizar colecciones existentes en ese idioma para ese usuario
<b>Autor</b>	Usuario
<b>Entradas</b>	Idioma
<b>Salidas</b>	Colecciones a visualizar
<b>Precondición</b>	Tener registradas colecciones en la base de datos
<b>Secuencia</b>	Seleccionar idioma Seleccionar botón colecciones Mostrar colecciones existentes en ese idioma
<b>Postcondición</b>	Página de visualización de colecciones

**Tabla 3.7:** *Caso de uso 07: Visualizar colecciones.*

<b>CU-08</b>	<b>Eliminar colección</b>
<b>Descripción</b>	Eliminar colección
<b>Autor</b>	Usuario
<b>Entradas</b>	Idioma
<b>Salidas</b>	Colecciones a eliminar
<b>Precondición</b>	-
<b>Secuencia</b>	Seleccionar colecciones a eliminar Mostrar diálogo de confirmación Eliminar colecciones en la base de datos Actualizar pantalla de visualización de colecciones
<b>Postcondición</b>	Colección eliminada

**Tabla 3.8:** *Caso de uso 08: Eliminar colección.*

<b>CU-09</b>	<b>Visualizar palabras de una colección</b>
<b>Descripción</b>	Visualizar palabras de una colección
<b>Autor</b>	Usuario
<b>Entradas</b>	Colección
<b>Salidas</b>	Palabras pertenecientes a la colección
<b>Precondición</b>	Tener registradas palabras en la colección
<b>Secuencia</b>	Seleccionar botón colecciones Seleccionar colección a visualizar Mostrar palabras existentes en esa colección
<b>Postcondición</b>	Página de visualización de palabras

**Tabla 3.9:** *Caso de uso 09: Visualizar palabras de una colección.*

<b>CU-10</b>	<b>Visualizar perfil</b>
<b>Descripción</b>	Visualizar perfil de usuario
<b>Autor</b>	Usuario
<b>Entradas</b>	Idioma
<b>Salidas</b>	Nombre de usuario Número de colecciones del usuario en ese idioma Número de palabras del usuario en ese idioma Últimas partidas jugadas en ese idioma
<b>Precondición</b>	-
<b>Secuencia</b>	Seleccionar idioma Seleccionar botón perfil Mostrar número de colecciones existentes en ese idioma Mostrar número de palabras existentes en ese idioma Mostrar últimas partidas jugadas en ese idioma
<b>Postcondición</b>	Página de perfil

**Tabla 3.10:** *Caso de uso 10: Visualizar perfil.*

<b>CU-11</b>	<b>Jugar a Test, El intruso o Desordenado</b>
<b>Descripción</b>	Jugar a uno de los 3 juegos mencionados
<b>Autor</b>	Usuario
<b>Entradas</b>	Colecciones seleccionadas para jugar
<b>Salidas</b>	Puntuación de la partida
<b>Precondición</b>	Tener registradas más de 10 palabras
<b>Secuencia</b>	Seleccionar tipo de juego Seleccionar colecciones con las que practicar Comprobar que el número de palabras seleccionadas es superior a 10 Crear juego con las palabras seleccionadas Jugar al juego Comprobar las vidas o el tiempo restante Una vez se hayan acabado, redirigir a la pantalla final Calcular puntuación dependiendo de las características de la partida Registrar partida en base de datos Mostrar puntuación en pantalla final
<b>Postcondición</b>	Página de visualización de partida

**Tabla 3.11:** *Caso de uso 11: Jugar a Test, El intruso o Desordenado.*

<b>CU-12</b>	<b>Jugar a Wordle</b>
<b>Descripción</b>	Jugar al juego Wordle
<b>Autor</b>	Usuario
<b>Entradas</b>	Idioma
<b>Salidas</b>	Resultado del intento
<b>Precondición</b>	Tener registrada alguna palabra de menos de 7 letras
<b>Secuencia</b>	Seleccionar tipo de juego Wordle Comprobar que haya una partida iniciada ese mismo día Si la hay, cargar los intentos y la solución de la base de datos Si no hay partida ese día, registrar una solución en la base de datos Si no se ha llegado a 5 intentos, cargar plantilla de nuevo intento Validar el intento añadido Comprobar la posición de las letras del intento y la solución Mostrar diferentes colores según la comprobación Comprobar que el intento añadido no era el quinto Comprobar que si se ha llegado a los 5 intentos Si se ha llegado a 5 intentos, mostrar mensaje informativo
<b>Postcondición</b>	Comprobación del intento en Wordle

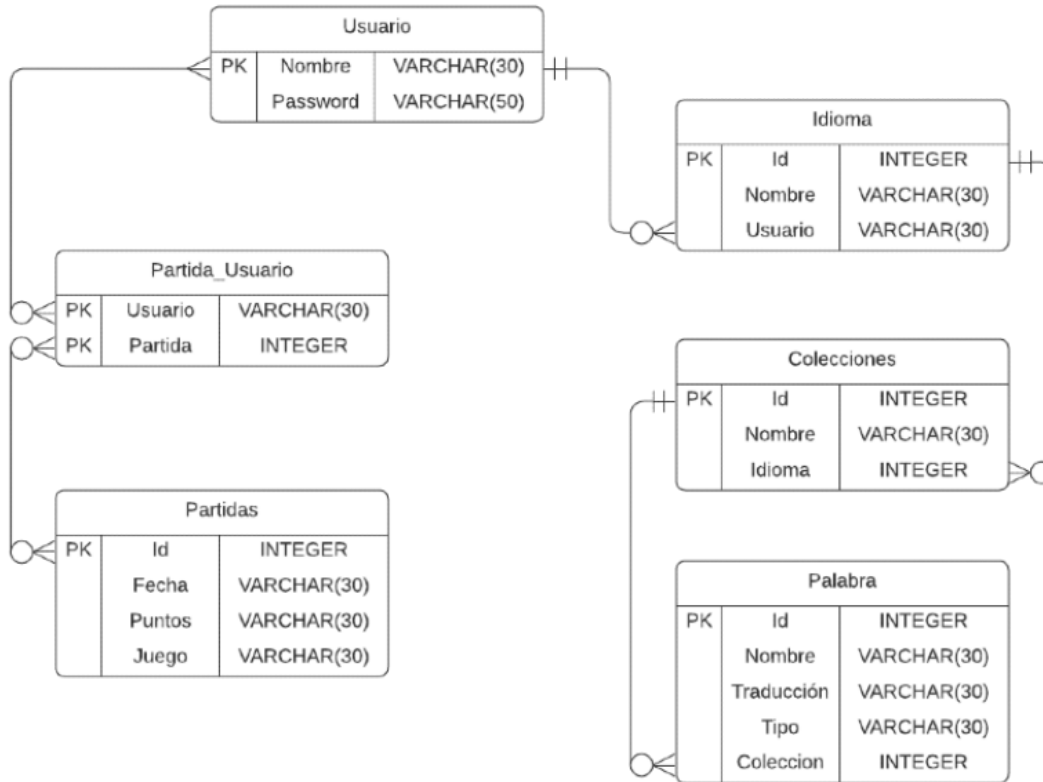
**Tabla 3.12:** *Caso de uso 12: Jugar a Wordle.*

### 3.4. Base de datos

Esta aplicación solo requiere el uso de una base de datos local, por lo que se ha optado por utilizar SQLite [13]. Este sistema de bases de datos es el ideal para dispositivos móviles, ya que se basa en ficheros y no requiere de un servicio adicional en ejecución.

Como se puede observar en la Figura 3.3, la base de datos está dividida en un total de 6 tablas de la siguiente forma:

- La tabla de usuarios, “Usuario”, donde están almacenados los nombres de usuario y sus contraseñas previamente cifradas.
- La tabla de idiomas, “Idioma”, donde están almacenados los identificadores de cada idioma, su nombre y el nombre de usuario al que corresponde.
- La tabla de colecciones, “Colecciones”, donde están almacenados los identificadores de cada colección, sus nombres y el identificador del idioma al que pertenecen.

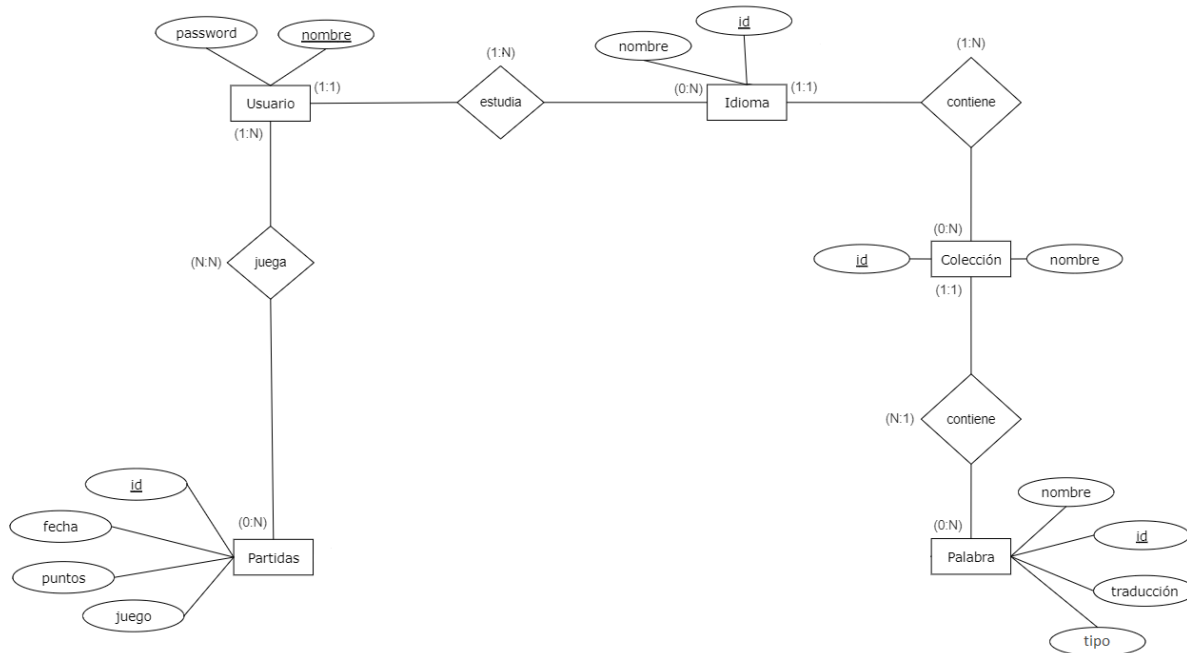


**Figura 3.3:** Modelo UML de la base de datos.

- La tabla de palabras, “Palabra”, donde están almacenados los identificadores de cada palabra, el nombre de la palabra, su traducción, el tipo de palabra y el identificador de la colección al que pertenece.
- La tabla de partidas, “Partidas”, donde están almacenados los identificadores de cada partida, la fecha en la que se realizaron, el tipo de juego y los puntos conseguidos.
- La tabla “Partida\_Usuario”, donde se relacionan las partidas realizadas con los usuarios que jugaron esa partida.

En estas tablas se ha utilizado el tipo *int* para los identificadores y *varchar* para los textos, ya que se ha determinado que ninguna palabra, traducción o colección iba a tener más de 30 caracteres de longitud.

La estructura relacional entre las tablas de la base de datos, que ofrecen la consistencia característica de este sistema de almacenamiento de datos y que se puede observar en la Figura 3.4, está formada por las siguientes 4 relaciones:



**Figura 3.4:** *Modelo Entidad-Relación de la base de datos.*

- 1-N entre “Usuario” e “Idioma” mediante el nombre de usuario. Siendo obligatorio que cada usuario tenga al menos un idioma que le corresponda.
- 1-N entre “Idioma” y “Colección” mediante el identificador del idioma. Siendo posible que existe algún idioma al que no le corresponda ninguna colección.
- 1-N entre “Colección” y “Palabra” mediante el identificador de la colección. Siendo posible la existencia de una colección al que no le corresponda ninguna palabra.
- N-N entre “Usuario” y “Partidas” mediante el identificador del usuario y el identificador de la partida. Siendo posible que la existencia de un usuario al que no le corresponda ninguna partida.

## 3.5. Seguridad

La única medida de seguridad a tomar en esta aplicación es la del cifrado de la contraseña de cada usuario para que nadie pueda acceder a ellas. Para conseguirlo se ha utilizado la API de android destinada al manejo de cifrados. Se ha optado por utilizar un algoritmo de cifrado SHA-512[14] ya que es uno de los más utilizados para proteger contraseñas en la actualidad.

Este algoritmo hash, proveniente del conjunto de algoritmos SHA-2, realiza 80 iteraciones con operadores lógicos al texto de entrada (en este caso la contraseña) y acaba resultando en una string de 512 bits, que es el que se almacena en la base de datos. Al iniciar sesión se vuelve a utilizar este algoritmo y se compara el resultado de esa función con la contraseña almacenada en la base de datos, de esta forma, la contraseña que el usuario introduce nunca se almacena, y por lo tanto, la aplicación resulta segura.

# Capítulo 4

## Implementación

### 4.1. Implementación de funcionalidades: Back-End

En esta sección se describirán algunos de los detalles del código utilizado para las funcionalidades de la aplicación.

#### 4.1.1. Inicio de sesión y registro

Todas las funcionalidades de la aplicación necesitan de un inicio de sesión previo por parte del usuario. Para poder almacenar el nombre de usuario y su contraseña se utiliza la API de Android destinada al manejo de SQLite, como podemos ver en la Figura 4.1. Además, también se comprueba si ya existe o no ese nombre de usuario y si no existe se almacena un nuevo registro en la tabla Usuarios, dándole permiso para utilizar el resto de la aplicación.

```
long idResultante = db.insert(Utilidades.TABLA_USUARIOS, Utilidades.USUARIOS_NOMBRE,
    user.toContentValues());
```

**Figura 4.1:** Código utilizado para registrar usuario.

#### 4.1.2. Introducción de palabras y colecciones

La funcionalidad principal de la aplicación es almacenar traducciones de palabras y organizarlas para poder practicar con algún grupo concreto. A continuación se explica cómo se ha realizado esta funcionalidad.

Para almacenar una palabra y su correspondiente traducción se ha creado un formulario XML. Como todos los campos se utilizan en los juegos disponibles para el usuario, es obligatorio rellenarlos, así que se hace una comprobación previa impidiendo al usuario registrar una palabra en la base de datos con un campo nulo. Además, para impedir que el usuario vuelva a almacenar la misma palabra en la misma colección, se hace uso de la API de Android destinada al manejo de SQLite para buscar las palabras de la colección seleccionada y comprobar que no va a haber un duplicado, como podemos ver en la Figura 4.2.

Esta misma query es la que se realiza a la hora de buscar las palabras pertenecientes a una misma colección.

```
String query = "SELECT * FROM " + Utilidades.TABLA_PALABRAS + " WHERE " +
    Utilidades.PALABRAS_COLECCION + "=?";
Long coleccionId = colecciones.get(spinner.getSelectedItemPosition()).getId();
String[] selectionArgs = new String[]{coleccionId.toString()};
Cursor c = db.rawQuery(query, selectionArgs);
```

**Figura 4.2:** Código utilizado para buscar las palabras pertenecientes a una colección.

### 4.1.3. Prácticas

La práctica *Test* se ha implementado realizando una búsqueda de todas las palabras incluidas previamente por el usuario a través de las colecciones seleccionadas. Estas se almacenan en un ArrayList que contiene todas las palabras y en otro que depende del tipo de palabra que sea.

Como hemos almacenado cada palabra en 2 listas independientes, a la hora de seleccionar las 3 opciones incorrectas podemos comprobar si el usuario tiene disponibles más de 4 palabras de ese tipo. Si no las tiene, las opciones incorrectas se seleccionarán aleatoriamente del conjunto global de las palabras añadidas. Si las tiene, se seleccionarán aleatoriamente dentro del subconjunto del tipo de palabra que se haya preguntado. De esta forma, el usuario tendrá más dificultades para encontrar la palabra correcta, ya que todas las opciones serán el mismo tipo de palabra. Podemos observar esta funcionalidad con mayor claridad en la Figura 4.3.

Además, cada vez que se selecciona una palabra como pregunta, esta se elimina de su correspondiente ArrayList, de modo que sea imposible volverla a preguntar en ese mismo juego.

```
if (sustantivos.size() < 4) {
    getPalabrasAleatorias(pre);
} else {
    pre.setOpcion2(sustantivos.get(rand.nextInt(sustantivos.size())).getTraduccion());
    while (pre.getOpcion2().equals(pre.getOpcion1()))
        pre.setOpcion2(sustantivos.get(rand.nextInt(sustantivos.size())).getTraduccion());
    pre.setOpcion3(sustantivos.get(rand.nextInt(sustantivos.size())).getTraduccion());
    while (pre.getOpcion3().equals(pre.getOpcion1()) || pre.getOpcion3().equals(pre.getOpcion2()))
        pre.setOpcion3(sustantivos.get(rand.nextInt(sustantivos.size())).getTraduccion());
    pre.setOpcion4(sustantivos.get(rand.nextInt(sustantivos.size())).getTraduccion());
    while (pre.getOpcion4().equals(pre.getOpcion1()) || pre.getOpcion4().equals(pre.getOpcion2())
        || pre.getOpcion4().equals(pre.getOpcion3()))
        pre.setOpcion4(sustantivos.get(rand.nextInt(sustantivos.size())).getTraduccion());
}
```

**Figura 4.3:** Código utilizado para seleccionar las distintas opciones.

La práctica *El intruso* se ha implementado reutilizando el código de la práctica Test, ya que su modo de juego es completamente similar. El único cambio realizado ha sido determinar cuándo el usuario no puede continuar jugando. En esta modalidad, si el usuario no tiene más de 4 palabras en ninguno de los ArrayLists de los tipos de palabra, su juego habrá terminado.

La práctica *Desordenado* se ha implementado, al igual que las 2 anteriores, incluyendo las palabras añadidas por el usuario en la selección de colecciones. El usuario dispone de un ProgressBar que va disminuyendo a medida que pasa el tiempo. Si el usuario acierta la palabra preguntada, este ProgressBar (y su correspondiente string que indica los segundos restantes) se ven incrementados.

Cada letra está implementada como un botón personalizado, en el que se incluyen funcionalidades añadidas. Una de ellas es incluir otro botón adjunto cada vez que se pulse sobre él. De esta forma, podremos tener un mejor control sobre la visualización de los botones de pregunta y de respuesta.

Además, en esta clase `ButtonDesordenado`, se implementa el patrón observador. De esta forma, cada vez que el usuario seleccione una letra para añadirla a la solución se comprobará si todos los huecos de la solución están rellenos o no. Si lo están, los botones cambiarán de color, se escuchará un audio determinando que es correcta, y tras 0.5 segundos (a través de un instancia `Handler`, como podemos observar en la Figura 4.4) se escogerá una nueva pregunta con sus correspondientes letras aleatorias y se incrementará el tiempo disponible. Si no lo están, los botones también cambiarán su color y se escuchará otro audio que determina que la solución es incorrecta, pero esta vez los botones volverán a sus respectivas posiciones en la tabla de letras.

```
AudioManager audio = (AudioManager) getSystemService(Context.AUDIO_SERVICE);
respuestaCorrecta.start();
correctasCount++;
for (int i = 0; i < answer.size(); i++) {
    answer.get(i).marcarCorrecta();
}
Handler handler = new Handler();
handler.postDelayed(new Runnable() {
    @Override
    public void run() {
        layout_ques.removeAllViews();
        layout_sol.removeAllViews();
        answer.clear();
        escogerPalabra();
        fillLetrasSolucion();
        asignarBotones();
    }
}, delayMillis: 500);
```

**Figura 4.4:** Código utilizado para marcar respuesta correcta.

El juego *Wordle* es completamente diferente a los otros 3 mencionados anteriormente. A diferencia de ellos, solamente se le permite al usuario practicarla un vez al día. Por lo que, nada más comenzar este juego, hace una búsqueda en la base de datos por si hubiese alguna partida registrada ese mismo día. Si lo hubiese, carga la solución almacenada anteriormente y los intentos que haya realizado el usuario. Si no la hubiese, se selecciona una palabra completamente aleatoria, de menos de 7 caracteres, que el usuario tenga almacenada.

Cada vez que el usuario registra un intento, se almacena en la base de datos, se reflejan los aciertos/fallos que haya podido obtener y se crea otro layout vacío a modo de nuevo intento. Para evitar despistes del usuario, solamente se registrarán intentos que tengan la misma cantidad de caracteres que la solución, si el usuario intenta registrar una palabra con un número diferente, aparecerá una alerta negándole el intento.

Por cada letra del intento del usuario se comprobará si: esa letra está presente en la solución pero se encuentra en otra posición, esa letra está presente en la solución en esa misma posición o si esa letra está presente en la solución en esa misma posición y además se repite en otra posición más. Esta comprobación la podemos observar en la Figura 4.5.

```
if (Character.toLowerCase(intento.charAt(i)) == Character.toLowerCase(solucion.charAt(i))) {  
    if (count == 1) {  
        b.setBackground(getDrawable(R.drawable.btn_desordenado_answer_completed));  
    }  
    else {  
        b.setBackground(getDrawable(R.drawable.btn_desordenado_answer_duplicated));  
    }  
} else {  
    if (count > 1) {  
        b.setBackground(getDrawable(R.drawable.btn_desordenado_answer_incomplete));  
    }  
    b.setBackground(getDrawable(R.drawable.btn_desordenado_question_error));  
}  
b.setText("" + intento.charAt(i));
```

**Figura 4.5:** Código utilizado para comprobar una letra en Wordle.

#### 4.1.4. Multilinguaje

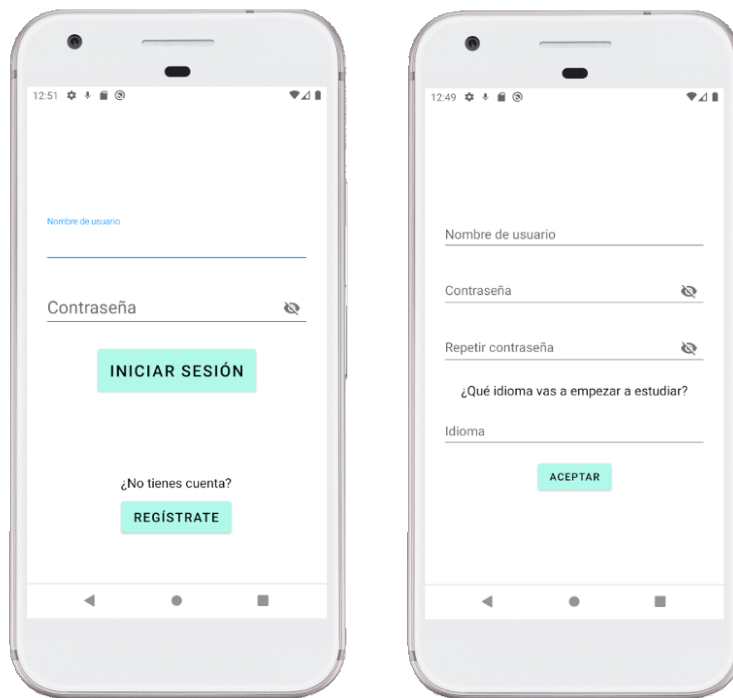
Por último, cabe destacar que se han utilizado archivos XML para la traducción de la aplicación. Si el dispositivo en el que se está ejecutando tiene configurado el idioma inglés, todos los textos que figuren en la aplicación se mostrarán en ese idioma. Si tiene configurado cualquier otro idioma, se mostrarán en español.

## 4.2. Visualización de las salidas: Front-End

Como existe la posibilidad de que distintos usuarios compartan el mismo dispositivo móvil y la premisa fundamental de la aplicación es poder tener las palabras y traducciones perfectamente ordenadas, la aplicación se empezó desarrollando las vistas de inicio de sesión y de registro.

Tal y como podemos observar en la Figura 4.6, la primera actividad consta de un formulario que permite rellenar el nombre de usuario y la contraseña para poder acceder a la aplicación. Además, existe un botón con el que poder navegar a la segunda vista.

Esta consta de un formulario con el nombre de usuario, su contraseña y el primer idioma que está dispuesto a estudiar (tal y como vemos en la figura 4.3). De esta forma, nada más registrarse puede acceder a la pantalla principal de la aplicación insertando los respectivos registros en las tablas Usuarios e Idiomas de la base de datos [15].



**Figura 4.6:** *Vistas de Inicio de sesión y registro.*

La pantalla principal de la aplicación es una vista la cual recoge 4 botones con los que

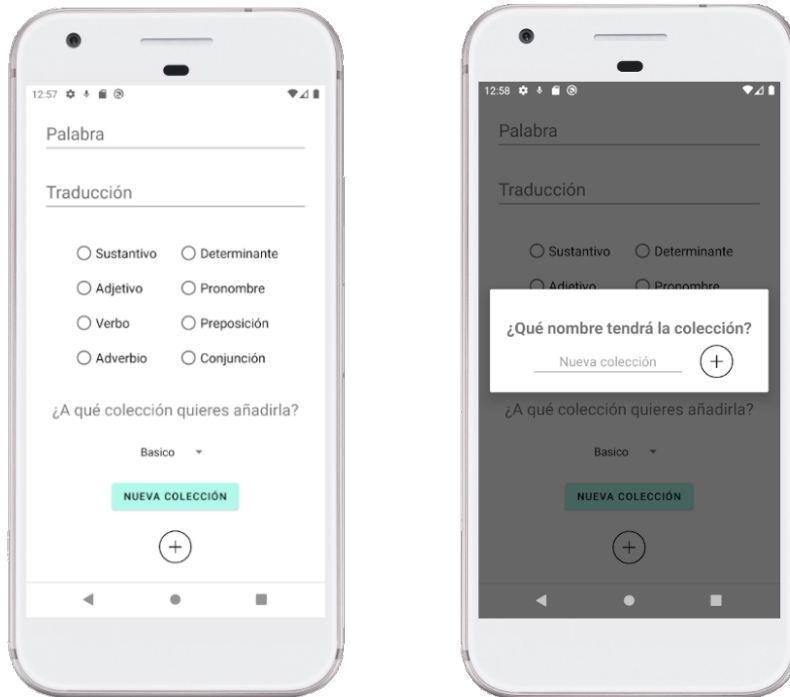
navegar a las funcionalidades principales de la aplicación: añadir una palabra, visualizar palabras de las colecciones existentes, practicar con los juegos disponibles y visualizar la página de perfil del usuario. Además, como podemos ver en la Figura 4.7, se añadió un TextView indicando el número de palabras almacenadas en ese idioma y dos botones adicionales que permiten al usuario cambiar de idioma o cambiar de usuario.



**Figura 4.7:** *Vista de pantalla principal.*

Para añadir una palabra se ha optado por crear un formulario simple en el que rellenar la misma palabra, su traducción, seleccionar el tipo de palabra y la colección a la que el usuario quiere añadirla. Tal y como podemos observar en la Figura 4.8, en esta vista se ha incluido un botón que permite al usuario crear una nueva colección. Al pulsar sobre él, se muestra un diálogo en el que el usuario puede decidir el nombre que tendrá la nueva colección.

La vista para visualizar las colecciones se compone de un RecyclerView al que se le ajunta un GridLayoutManager. Como podemos observar en la Figura 4.6, de esta forma se ha conseguido obtener una especie de tabla en el que cada ítem es un botón con el nombre de cada colección adjunto. Al pulsar sobre cada botón, podemos navegar a otra



**Figura 4.8:** *Vista de añadir una palabra.*

pantalla en la que se muestran todas las palabras que forman esa colección. Sin embargo, si mantenemos pulsado el botón de una colección, aparecerá una ventana de diálogo para confirmar si queremos eliminar esa colección, tal y como se muestra en la Figura 4.9.

La lista de palabras de cada colección consta de un RecyclerView, esta vez con un LinearLayoutManager adjunto, que aloja tantos ítems como palabras tenga la colección. Estos ítems (mostrados en la Figura 4.10) están formados por un CardView, que a su vez está formado por 3 TextViews (que muestran la palabra, su traducción y el tipo de palabra) y 2 botones. Estos últimos son los que permiten al usuario realizar modificaciones.

El primero direcciona al usuario a la pantalla de Añadir Palabra con ligeras modificaciones. En vez de tener todos los campos vacíos, estos son rellenados con los datos de la palabra a editar para poder modificar únicamente los campos que el usuario necesite. Además, el botón para añadir una colección se vuelve invisible (puesto que la modificación de los datos de una palabra no requiere de ninguna colección nueva) y no se podrá modificar la colección en la que está ubicada la palabra.

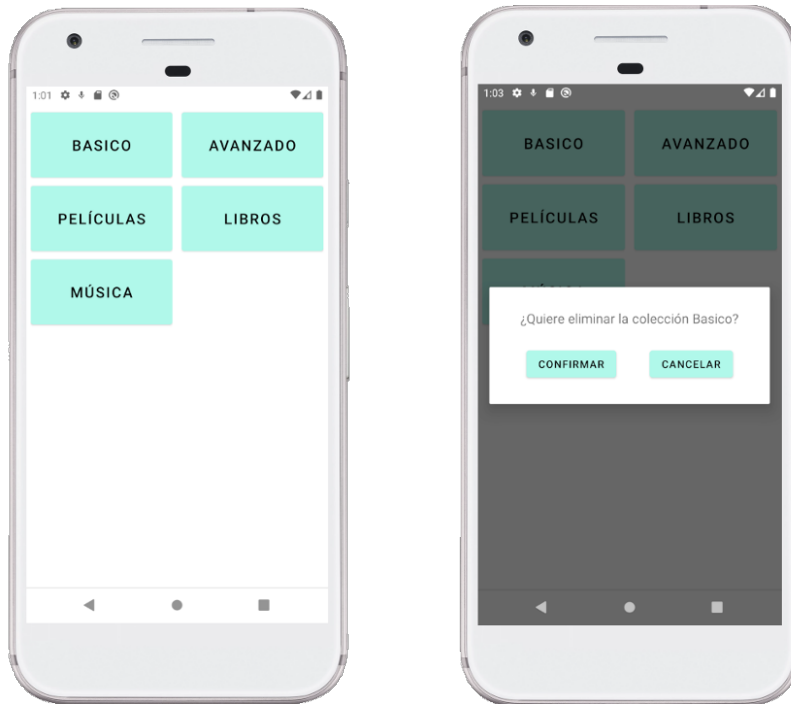


Figura 4.9: Vista de listado de colecciones.

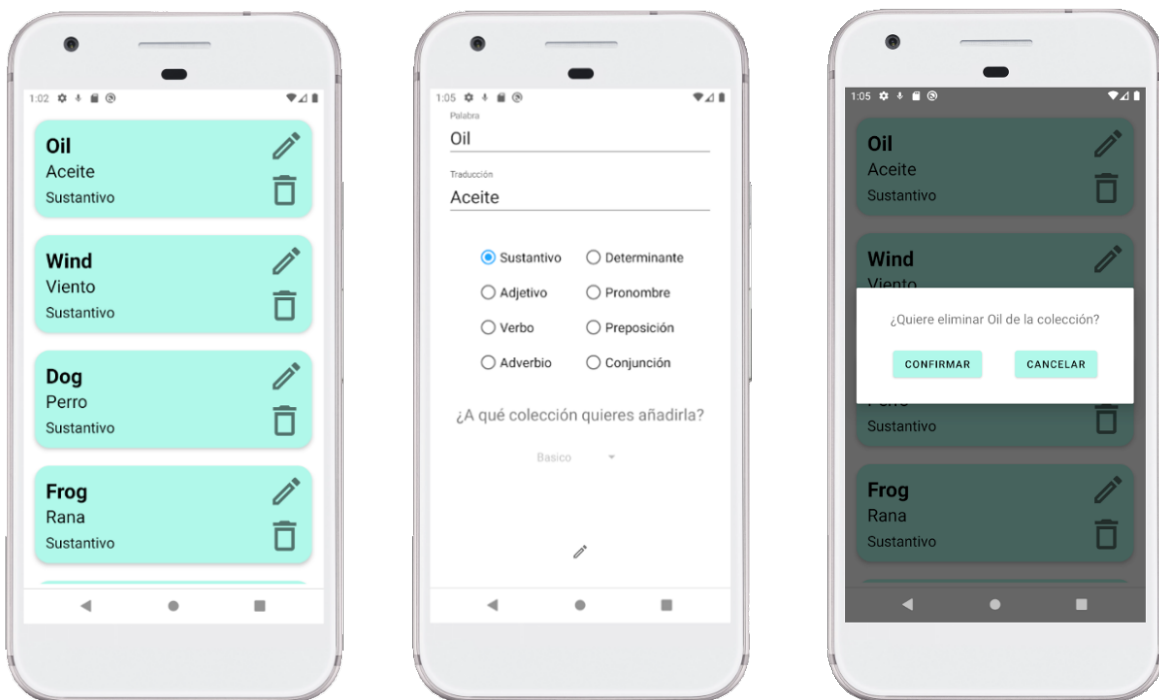


Figura 4.10: Vista de listado de palabras.

El segundo hace visible un dialogo de confirmación en el que se comprueba que el usuario realmente quiere eliminar esa palabra de la colección.

Para seleccionar la práctica a realizar se optó por crear una simple vista con 4 botones que determinan el tipo de juego. Los 3 primeros botones direccionan al usuario a otra vista en la que deberá incluir las colecciones que quiera practicar en ese momento. Esta vista incluye un TextView que muestra el número de palabras añadidas dinámicamente y un RecyclerView que aloja en forma de tabla todas las colecciones almacenadas por el usuario en ese idioma. Además, como nos muestra la Figura 4.11, se ha añadido un botón debajo de la lista de colecciones que permite al usuario comenzar la práctica.

Las vistas de cada práctica se podrán visualizar en el apéndice Flujo de uso y Demo de la aplicación.



**Figura 4.11:** *Vista de selección de práctica.*

La vista del perfil se comprende de 4 TextViews que ofrecen datos del usuario (nombre de usuario, el idioma en el que se encuentra y el número de colecciones y palabras que almacena en ese idioma). Además, tal y como observamos en la Figura 4.12, el usuario dispone de un

RecyclerView con sus últimas partidas (ordenadas por fecha) en la que podrá visualizar por medio de CardViews, la fecha en la que realizó esa práctica, el tipo de juego que fue y la cantidad de puntos que obtuvo.



**Figura 4.12:** *Vista de perfil.*



# Capítulo 5

## Resultados

### 5.1. Profiling o resultados técnicos

En esta sección se analiza el rendimiento de la aplicación. Para poder llevar a cabo este análisis, se ha utilizado la herramienta de creación de perfiles de la plataforma Android Studio. Las mediciones se han tomado usando un Xiaomi POCO X3 NFC, dispositivo con 6Gb de RAM, un procesador Qualcomm Snapdragon 732G y ejecutando la versión de Android 10 con la capa de personalización MIUI 12.0.8.

Los datos a medir son el tiempo de carga, que es el tiempo que tarda el dispositivo en realizar la consulta en la base de datos; el máximo uso de la CPU, que es el porcentaje de carga máxima a la que se encuentra la CPU al procesar la consulta; el uso de la memoria RAM, que es la cantidad de RAM consumida al procesar dicha consulta. Hay que tener en cuenta que los datos obtenidos pueden variar por muchos factores, como la versión del sistema operativo o el dispositivo utilizado. Por tanto, las mediciones pueden no coincidir exactamente con las mostradas a continuación.

#### **Añadir palabra**

Los resultados obtenidos en las pruebas de creación o registro de una palabra se pueden observar en la Tabla [5.1](#).

<b>Tiempo de carga</b>	114ms
<b>Uso max. CPU</b>	17 %
<b>Uso max. RAM</b>	112MB

**Tabla 5.1:** *Medidas de rendimiento en la creación de una palabra.*

### **Editar palabra**

Los resultados obtenidos en las pruebas de modificación de una palabra registrada se pueden observar en la Tabla 5.2.

<b>Tiempo de carga</b>	87ms
<b>Uso max. CPU</b>	14 %
<b>Uso max. RAM</b>	100MB

**Tabla 5.2:** *Medidas de rendimiento en la edición de una palabra.*

### **Eliminar palabra**

Los resultados obtenidos en las pruebas de eliminación de una palabra registrada se pueden observar en la Tabla 5.3.

<b>Tiempo de carga</b>	114ms
<b>Uso max. CPU</b>	12 %
<b>Uso max. RAM</b>	82MB

**Tabla 5.3:** *Medidas de rendimiento en la eliminación de una palabra.*

### **Crear colección**

Los resultados obtenidos en las pruebas de creación o registro de una colección se pueden observar en la Tabla 5.4.

<b>Tiempo de carga</b>	53ms
<b>Uso max. CPU</b>	3 %
<b>Uso max. RAM</b>	76MB

**Tabla 5.4:** *Medidas de rendimiento en la creación de una colección.*

### Eliminar colección

Los resultados obtenidos en las pruebas de eliminación de una colección registrada se pueden observar en la Tabla 5.5.

Tiempo de carga	114ms
Uso max. CPU	13 %
Uso max. RAM	112MB

**Tabla 5.5:** *Medidas de rendimiento en la eliminación de una colección.*

### Listar palabras

Los resultados obtenidos en las pruebas de listado de las palabras registradas en una colección se pueden observar en la Tabla 5.6.

Tiempo de carga	59ms
Uso max. CPU	10 %
Uso max. RAM	100MB

**Tabla 5.6:** *Medidas de rendimiento en el listado de las palabras.*

### Listar colecciones

Los resultados obtenidos en las pruebas de listado de las colecciones registradas en un idioma se pueden observar en la Tabla 5.7.

Tiempo de carga	31ms
Uso max. CPU	3 %
Uso max. RAM	95MB

**Tabla 5.7:** *Medidas de rendimiento en el listado de las colecciones.*

### Registrar usuario

Los resultados obtenidos en las pruebas de registro de usuario se pueden observar en la Tabla 5.8.

Tiempo de carga	48ms
Uso max. CPU	15 %
Uso max. RAM	100MB

**Tabla 5.8:** *Medidas de rendimiento en el registro de un usuario.*

## 5.2. Resultados de usuarios

Dado que esta aplicación se centró en ser accesible para el usuario final, se ha realizado una encuesta a las personas que han tenido acceso a ella. Después de utilizar la aplicación han tenido que responder a una serie de preguntas. Están divididas en 2 tipos: un formulario valorando del 0 al 10, siendo 0 totalmente en desacuerdo y 10 totalmente de acuerdo; y preguntas de desarrollo. Las preguntas fueron las siguientes:

### **Formulario**

1. Usaría esta aplicación frecuentemente.
2. Es complejo navegar entre las pantallas de la aplicación.
3. La aplicación es fácil de usar.
4. Las funciones están bien integradas.
5. La aplicación es muy inconsistente.
6. La mayoría de la gente aprendería a utilizar esta aplicación de forma rápida.
7. Las dinámicas de los juegos se entienden bien.
8. Algunas funcionalidades son incómodas a la hora de realizarlas.

### **Cuestiones**

1. ¿Te sientes incómodo teniendo que navegar hasta el menú de añadir palabra para añadir una colección?
2. ¿Qué es lo que menos te gusta de la aplicación?
3. ¿Echas en falta alguna funcionalidad?

### Usuario 1: Helga.

Estudiante de Administración y Dirección de Empresas y Derecho. 22 años.

### Formulario

En la Tabla 5.9 se pueden observar las respuestas ofrecidas por Helga a las cuestiones del formulario.

Número de pregunta	Valoración
1	8
2	2
3	9
4	9
5	0
6	8
7	10
8	2

**Tabla 5.9:** *Valoración de Helga.*

### Cuestiones

1. No.
2. Creo que el juego de Wordle es demasiado complicado si el usuario tuviese una colección de palabras significativamente grande. Desde mi punto de vista, sería más sencillo si diesen la letra inicial o alguna de las letras integrantes en la palabra como pista.
3. Más variedad de juegos o poder interactuar con otros usuarios que tengan la aplicación. Además, estaría bien una leyenda con la que entender los colores de Wordle.

**Usuario 2: David.**

Ingeniero industrial. 24 años.

**Formulario**

En la Tabla 5.10 se pueden observar las respuestas ofrecidas por David a las cuestiones del formulario.

Número de pregunta	Valoración
1	9
2	1
3	7
4	8
5	1
6	9
7	10
8	1

**Tabla 5.10:** *Valoración de David.*

**Cuestiones**

1. No.
2. La práctica Desordenado se me hace demasiado complicada.
3. La búsqueda de palabras.

### Usuario 3: Javier.

Ingeniero informático. 32 años.

### Formulario

En la Tabla 5.9 se pueden observar las respuestas ofrecidas por Javier a las cuestiones del formulario.

Número de pregunta	Valoración
1	9
2	3
3	8
4	8
5	0
6	9
7	8
8	1

**Tabla 5.11:** *Valoración de Javier.*

### Cuestiones

1. No.
2. El diseño de la interfaz.
3. Poder seleccionar las últimas palabras añadidas a la hora de practicar (dentro de la misma colección).

#### Usuario 4: Diego.

Ingeniero de las energías. 25 años.

#### Formulario

En la Tabla 5.12 se pueden observar las respuestas ofrecidas por Diego a las cuestiones del formulario.

Número de pregunta	Valoración
1	9
2	2
3	8
4	9
5	1
6	10
7	9
8	1

**Tabla 5.12:** *Valoración de Diego.*

#### Cuestiones

1. No.
2. El tener que añadir palabras una por una.
3. Poder enviar colecciones agregadas entre amigos.

#### Análisis de las valoraciones de los usuarios encuestados

La mayoría de los usuarios han encontrado esta aplicación bastante utilizable. Además, están de acuerdo en que es bastante fácil navegar por la interfaz y realizar las distintas funcionalidades que se ofrecen.

Por último, muchos han querido aportar ideas que podrían ser interesantes a la hora de continuar con el desarrollo de la aplicación. Algunas de estas se han incluido en la Sección 6.2.

# Capítulo 6

## Conclusiones y trabajo futuro

Por último, como se mencionó en la Sección 1.4, en este capítulo se manifiesta la experiencia y los objetivos desarrollados durante la realización de este proyecto, además de algunas ideas que no han podido ser desarrolladas.

### 6.1. Conclusiones

Tras analizar el desarrollo del proyecto, es hora de realizar una valoración de los objetivos cumplidos y del trabajo realizado.

El objetivo de este proyecto era elaborar una aplicación móvil Android para dar soporte al aprendizaje de idiomas. Este objetivo podría marcarse como cumplido, ya que después del trabajo realizado se ha obtenido una aplicación sencilla y atractiva en la cual poder organizar palabras y sus respectivas traducciones en diferentes idiomas.

Además, la aplicación cuenta con 4 juegos con los que poder practicar las palabras introducidas. Esta opción puede resultar bastante atractiva para el usuario final y definitivamente ayuda a cumplir el objetivo marcado, puesto que es un gran soporte para repasar esas palabras descubiertas que se van anotando.

Gran parte del desarrollo de este proyecto ha sido gracias a asignaturas estudiadas a lo largo de la carrera, como pueden ser: Tecnología de la Programación, en la que se enseñan las nociones básicas de la programación orientada a objetos y se obtiene un conocimiento profundo de Java; Bases de Datos y Ampliación de Bases de Datos, en las que se enseñan y

asientan las bases de los sistemas de bases de datos; y Programación de Aplicaciones para Dispositivos Móviles, en la que se enseñan los primeros pasos a la hora de iniciar el desarrollo de una aplicación Android.

## 6.2. Trabajo futuro

En esta sección se incluyen diferentes ideas que podrían aumentar el valor del proyecto, pero que por complejidad y por falta de tiempo no ha sido posible desarrollarlas.

- Posibilidad de que varios usuarios puedan conectarse a una base de datos en tiempo real y practicar los juegos disponibles con las palabras que seleccione cada uno de ellos. Esto incrementaría la motivación de los usuarios a la hora de seguir practicando su nuevo idioma y les ayudaría a descubrir palabras que desconocían, ya que todos los usuarios aportarían sus propias colecciones.
- Incluir más variedad de juegos para practicar las palabras añadidas. Algunos de ellos utilizando otras tecnologías como el traductor de Google para poder practicar fonéticamente las palabras.
- Posibilidad de extraer e importar colecciones entre usuarios. Para que, por ejemplo, los profesores puedan compartir con sus alumnos las colecciones directamente y así los usuarios no tengan que incluir las palabras en la colección una a una.
- Posibilidad de filtrar las palabras a practicar por la fecha en la que fueron añadidas. De esta forma el usuario podría practicar con las últimas palabras que añadió y no tener que volver a recordar las palabras que ha estudiado una y otra vez.
- Añadir un apartado de estadísticas para el usuario, con el fin de que pueda visualizar cuáles son las palabras que más falla en los juegos o un porcentaje de acierto de las palabras de cada colección.

- Añadir notificaciones que avisen al usuario si la aplicación detecta que lleva sin utilizarse una cierta cantidad de tiempo. De esta forma el usuario no descuidaría su aprendizaje del idioma y se fomentaría el uso de la aplicación.
- Añadir un examen semanal o mensual en el que el usuario pueda practicar las palabras que más ha errado durante ese periodo de tiempo.
- Añadir una barra de búsqueda en la pantalla de colecciones para que el usuario pueda buscar directamente los detalles de una palabra en concreto.
- Además, es importante recordar las tareas que deben aparecer siempre en un proyecto software si se desea que siga activo a lo largo del tiempo.
  - Reescribir el código o parte de este con el fin de aplicar algoritmos más eficientes, entre otras cosas.
  - Actualizar las tecnologías utilizadas, con la finalidad de obtener una aplicación completamente segura independientemente del sistema.
  - Implementar nuevas funcionalidades con las que poder ampliar el mercado y seguir interesando a los usuarios recurrentes.



# Bibliografía

- [1] Carlos Luis Sánchez. *Gamificación en la educación: hacia una pedagogía para involucrar y motivar a los estudiantes*. Editorial Académica Española, 2020.
- [2] Carys Betteley. ¿Pueden apps como Duolingo o Babbel reemplazar las clases de idiomas?, 2020. <https://cybercrew.uk/software/app-store-vs-play-store/>. [Online; accedido el 30 de mayo de 2022].
- [3] Sara García Alcaide. *La Gamificación en la enseñanza-aprendizaje del Inglés como LE: Cómo nos ayuda la tecnología a aprender el idioma en el aula*. Editorial Académica Española, 2018.
- [4] Alberto Payo. Duolingo fue la app de idiomas más descargada en 2018, 2019. <https://cybercrew.uk/software/app-store-vs-play-store/>. [Online; accedido el 30 de mayo de 2022].
- [5] Sanja Zdraveska. Apple App Store vs Google Play Store (2022 Comparison), 2022. <https://cybercrew.uk/software/app-store-vs-play-store/>. [Online; accedido el 30 de mayo de 2022].
- [6] Luca Ardito, Riccardo Coppola, Giovanni Malnati, and Marco Torchiano. Effectiveness of kotlin vs. java in android app development tasks. *Information and Software Technology*, 127:106374, 2020.
- [7] Zbigniew M. Sikora. Introduction. In Zbigniew M. Sikora, editor, *Java*, The Morgan Kaufmann Series in Data Management Systems, pages 1–6. Morgan Kaufmann, San Francisco, 2003.

- [8] Frank Ableson, Robi Sen, and Chris King. *Android. Guía para desarrolladores*. Anaya Multimedia, 2011.
- [9] Mark L. Murphy. *The Busy Coder's Guide to Android Development*. CommonsWare, LLC, 2009.
- [10] Reto Meier. *Professional Android 4 Application Development*. John Wiley Sons, 2012.
- [11] Jules Wang. Google's latest Android version distribution numbers show 11 in dead heat with 10, 2021. <https://cybercrew.uk/software/app-store-vs-play-store/>. [Online; accedido el 30 de mayo de 2022].
- [12] Ruslan Malogulko. Easy patterns: Observer, 2018. <https://itnext.io/easy-patterns-observer-63c832d41ffd>.
- [13] Kunjal Panchal. Everything you need to know about SQLite Mobile database, 2019. <https://ourcodeworld.com/articles/read/737/everything-you-need-to-know-about-sqlite-mobile-database>.
- [14] Zaid Khaishagi. Cryptography: Explaining SHA-512, 2019. <https://medium.com/@zaid960928/cryptography-explaining-sha-512-ad896365a0c1>. [Online; accedido el 30 de mayo de 2022].
- [15] Kyle Mew. *Learning Material Design*. Packt Publishing, 2015.

# Apéndice A

## Introduction

### A.1. Motivation

Maintaining concentration in a classroom is often difficult, especially when it comes to introducing yourself to a language that is new to the learner. Usually it is frustrating to have no understanding of the vocabulary and to stop paying attention to the teacher and even abandoning the proposed goal.

The use of games in education has always been used to improve student concentration [1], to the extent that, with the advance of technology, many people have stopped attending academies (or, as in this case, language schools) thinking that a mobile app or a website will be able to reach the same goal [2].

However, the presence of a teacher is still very important for learning, since he or she can understand how the learner feels, recognize his or her strengths and weaknesses and act accordingly to eliminate his or her shortcomings.

This project values the cooperation of teachers with new technologies, developing a tool that allows them to distribute the most important vocabulary for each person and to encourage their students to take the initiative in further developing their learning on their own [3].

In addition, it is also designed for people who want to maintain a self-taught environment. They might organize the words they discover in their day-to-day lives: reading books,

watching movies, listening to music, among other activities. This application allows you to write down these words and practice them anywhere. For this reason, it was determined that LiMob was a great name, since “Li” comes from Linguae and Libello (Latin “Languages” and “Notebook”) and “Mob” comes from Mobile (English “Mobile”).

## A.2. Objectives

The main goal of the project is to develop an app for mobile devices with Android operating system. This application will allow its users to organize word translations and practice with them through games. For this purpose, specific functions defined in advance will be developed. These objectives include:

- Development of an application for mobile devices with Android operating system.
- Development and implementation of a local database for each user to store all the information related to their words and some details of their latest practices.
- Development and implementation through Java and XML of interactive games where you can practice the words stored in the database.
- Usage of the integrated development environment Android Studio, which will use the Java programming language to make an object-oriented design suitable for our needs.

## A.3. Work plan

To begin with, research was carried out on existing platforms dedicated to support language learning. To do this, most of the outstanding applications were downloaded and usage tests were conducted, observing their functionalities. Research was also carried out on the different options for developing Android apps.

The next step was to clarify the functionalities that the application should have and the data that should be used to make the usage of the application as user-friendly as possible. In

addition, a study was carried out on the two languages used (Java and XML) which helped to start the development.

In order to be able to progress consistently with the project, the database was designed and a mockup was made on which the bases that the interface of the final project would have been based.

What took up most of the time was the implementation of the application. Adding some details that were not considered in the previous phases, but that as the development progressed, it was noted that they could be a great help to the user. The code is available in the following link<sup>1</sup>.

Finally, it was concluded with the development of this report, in order to reflect the entire development of the project and to record the decisions made during the work.

By observing Figure A.1, we can understand, thanks to a Gantt diagram, the amount of time spent on each of the development blocks. Salmon colour was used to highlight activities related to the research of the tools for the future development of the project, blue colour for those related to the specification of the project, yellow colour for those related to the design, pink colour for those related to the implementation and grey colour for the final report.

## A.4. Organization of this report

This section describes the content and organisation of the sections of the document:

This section describes the content and organisation of the sections of the document:

- **Chapter 2:** Contains a market survey of the most relevant applications available for download at this time, which have features and natures similar to LiMob, as well as an analysis of the results observed after the survey. Continued by a brief description of the technologies used in the development of LiMob, together with reasons for their choice.

---

<sup>1</sup><https://github.com/AlvaroSan98/TFGIIdiomas>

# Plan de trabajo

2021/2022

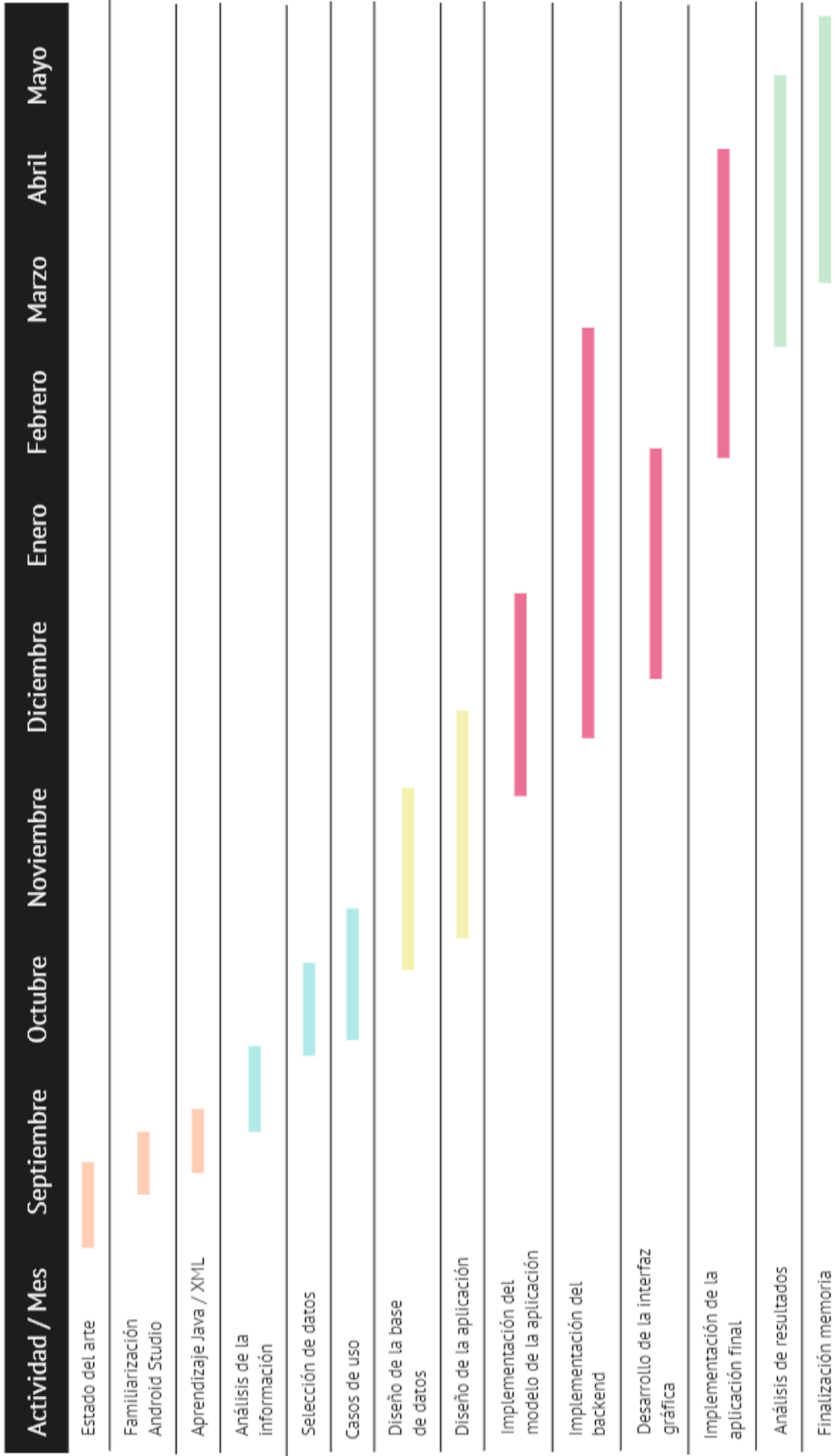


Figura A.1: Gantt Diagram.

- **Chapter 3:** Some of the most relevant use cases that users can perform in the application are presented, followed by the database model used, and the architecture that the application presents.
- **Chapter 4:** Details the entire development of the application, the design of the user interface and the components added to its different screens.
- **Chapter 5:** This is an analysis of the metrics obtained from the use of the app and feedback received from users who have been exposed to it.
- **Chapter 6:** Discusses the conclusions obtained during the development of this project and possible future work to enhance, augment or iterate functionality.
- Finally there is the bibliography and appendices which include a user guide and a demo video.



# Apéndice B

## Conclusions and future work

Finally, as mentioned in Section 1.4, this chapter presents the experience and objectives developed during the implementation of this project, as well as some ideas that could not be developed.

### B.1. Conclusions

After analysing the development of the project, it is time to make an assessment of the objectives achieved and the work carried out.

The aim of this project was to develop an Android mobile application to support language learning. This objective could be marked as achieved, since after the work carried out, a simple and attractive application has been obtained in which to organise words and their respective translations in different languages.

In addition, the application has 4 games with which to practice the words introduced. This option can be quite attractive for the end user and definitely helps to achieve the objective set, since it is a great support to review those words discovered that are being written down.

Much of the development of this project has been completed thanks to subjects studied throughout the degree, such as: Tecnología de la Programación, which teaches the basics of object-oriented programming and provides an in-depth knowledge of Java; Bases de Datos and Ampliación de Bases de Datos, which teaches and establishes the foundations of data-

base systems; and Programación de Aplicaciones para Dispositivos Móviles, which teaches the first steps when starting the development of an Android application.

## B.2. Future work

This section includes different ideas that could increase the value of the project, but due to complexity and lack of time it has not been possible to develop them.

- Possibility that several users can connect to a real time database and practice the available games with the words selected by each one of them. This would increase users' motivation to continue practising their new language and help them discover words they did not know, as all users would contribute their own collections.
- Include more variety of games to practice the added words. Some of them using other technologies such as Google translator to be able to practise the words phonetically.
- Possibility to extract and import collections between users. So that, for example, teachers can share the collections directly with their students so that users do not have to include the words in the collection one by one.
- Possibility to filter the words to be practised by the date they were added. This way the user could practice with the last words he/she added and not have to remember the words he/she has studied over and over again.
- Add a statistics section for the user, so that he/she can visualise which are the words he/she fails the most in the games or a percentage of success of the words in each collection. notifications that alert the user if the application detects that it has not been used for a certain amount of time. In this way the user would not neglect their language learning and the use of the application would be encouraged.
- Add a weekly or monthly test in which the user can practice the words he/she has missed the most during that period of time.

- Add a search bar on the collections screen so that the user can search directly for the details of a particular word.
- In addition, it is important to remember the tasks that must always appear in a software project if it is to remain active over time.
  - Rewrite the code or parts of it in order to implement more efficient algorithms, among other things.
  - Updating the technologies used, with the aim of obtaining a completely secure application regardless of the system.
  - Implement new functionalities with which to expand the market and continue to interest recurring users.



# Apéndice C

## Flujo de uso y Demo de la aplicación

En este anexo se va a mostrar el funcionamiento completo de las prácticas de la aplicación a través de un flujo de uso y una demostración de uso en vídeo.

### C.1. Flujo de uso

En esta sección se va a simular el flujo de uso de todas las practicas disponibles por parte de un usuario, mostrando las peculiaridades de cada una.

#### C.1.1. Test

Como vimos en la explicación del Front-End de la aplicación (concretamente en la Figura 4.11) antes de comenzar con este juego el usuario debería de haber seleccionado las colecciones que querría practicar. Una vez seleccionadas, comenzará el juego, y aparecerá una de las palabras aleatorias que el sistema ha seleccionado junto con 4 posibles opciones para el usuario.

Tal y como podemos observar en la Figura C.1, si el usuario selecciona una opción incorrecta aparecerá un símbolo de error debajo de las 4 opciones y adicionalmente se escuchará un sonido característico que simboliza que el usuario se ha equivocado. Sin embargo, si el usuario selecciona la opción correcta, la imagen mostrada y el sonido reproducido simbolizarán ese acierto. Cabe destacar, que si el usuario selecciona una opción incorrecta, las vidas del usuario bajarán, y esto será visible en la siguiente pregunta que tenga que realizar.

Una vez que el usuario se haya quedado sin vidas o haya respondido a todas las palabras seleccionadas, aparecerá una pantalla final que calculará los puntos obtenidos en esa práctica. El sistema de puntuación toma en cuenta la valentía del usuario a la hora de realizar la práctica, de esta forma, se premiará que seleccione muchas palabras antes de comenzar. Además, si el usuario ha respondido todas las palabras seleccionadas correctamente y le sobran vidas, también se tomará en cuenta a la hora de calcular su puntuación.

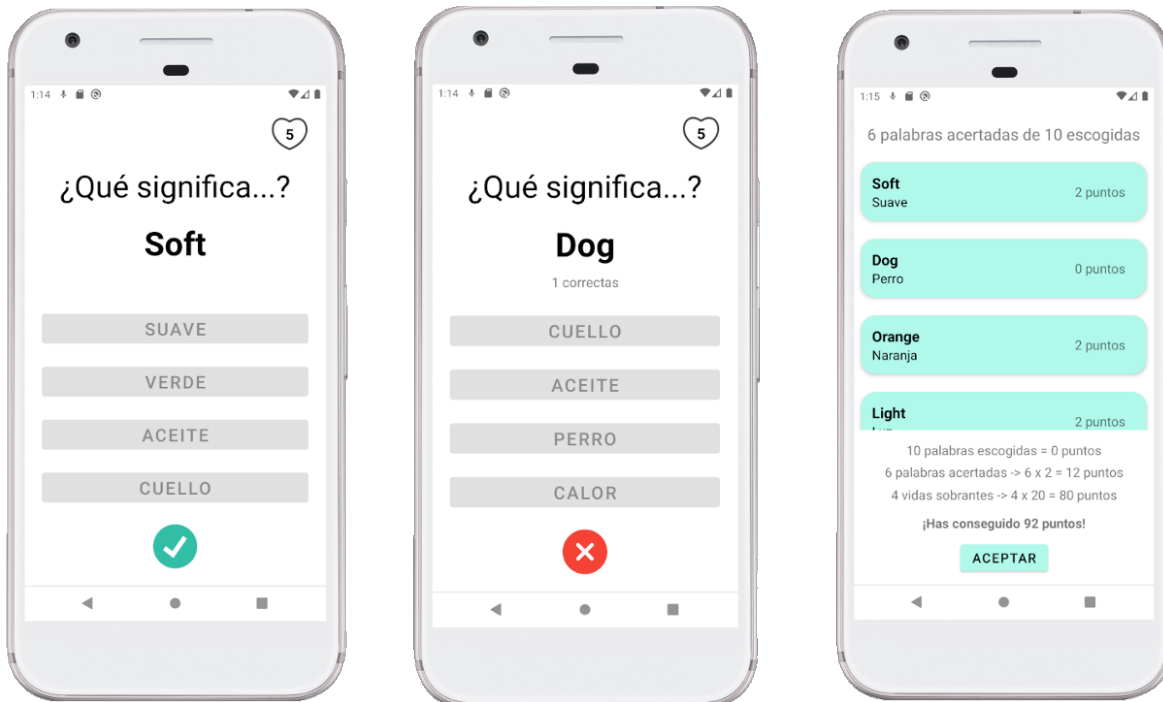


Figura C.1: Flujo de uso de la práctica Test.

### C.1.2. El intruso

Al igual que en la práctica anterior, el usuario deberá seleccionar las palabras a practicar antes de comenzar. Una vez seleccionadas, aparecerá un tipo de palabra junto con 4 opciones distintas. Estas opciones se corresponderán con 3 palabras que pertenezcan a ese tipo de palabra y otra que no. El usuario deberá seleccionar la palabra diferente a las otras 3.

Tal y como ocurría en la practica *Test*, si el usuario selecciona una palabra incorrecta, aparecerá una imagen y se reproducirá un sonido característico, y si la palabra seleccionada

es correcta, aparecerá otra imagen y se reproducirá el sonido correspondiente.

Cuando el usuario se haya quedado sin vidas o no haya más de 3 palabras en ninguno de los tipos de palabra disponibles (de esta forma sería imposible realizar esta práctica, ya que no se dispondría de 3 palabras similares para ser diferenciadas) el juego finalizará y aparecerá una pantalla final que calculará la puntuación de la práctica. Al igual que en la práctica anterior, se premiará la cantidad de palabras que el usuario haya seleccionado y las vidas restantes. Sin embargo, esta vez no podrá aparecer la lista de palabras y sus traducciones correspondientes junto con sus puntos, ya que en esta práctica las traducciones son irrelevantes. En este caso, la aplicación dispone de 3 frases distintas para mostrar según el porcentaje de acierto del usuario: “¡Buen intento!”, “Se puede mejorar...” o “¡Ponte las pilas!”. Esta pantalla se puede observar en la Figura C.2.

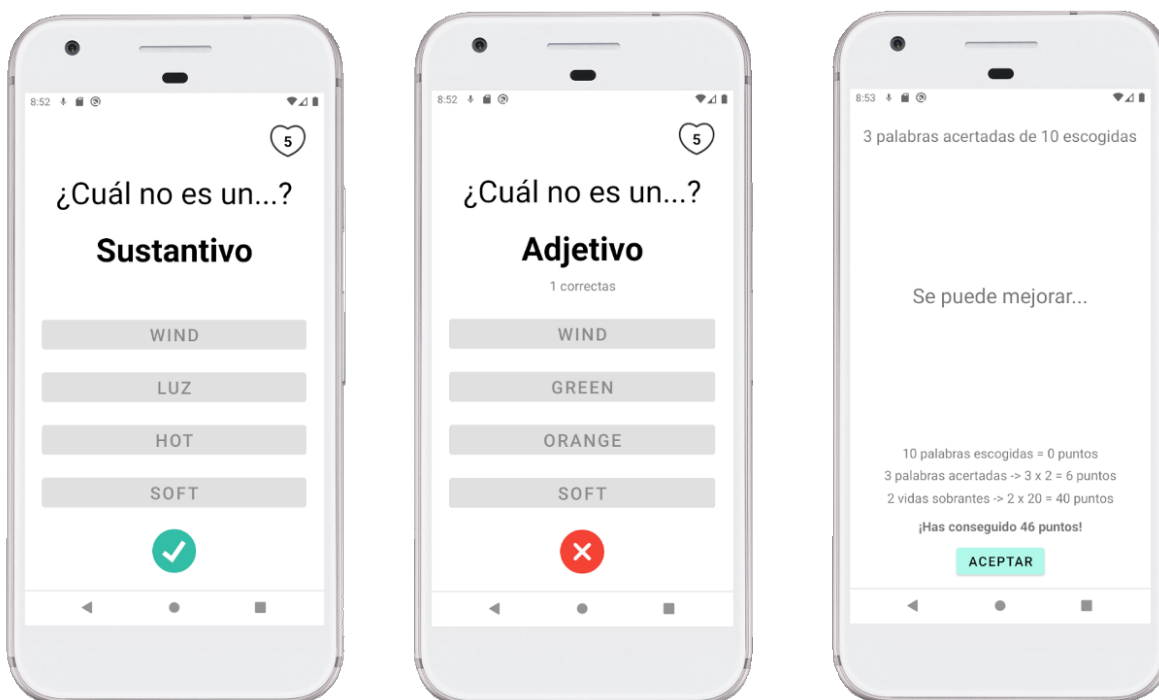


Figura C.2: Flujo de uso de la práctica Intruso.

### C.1.3. Desordenado

De la misma forma que en las prácticas anteriores, para que esta práctica se pueda realizar el usuario deberá de haber seleccionado las colecciones que desee practicar. Una vez el juego comienza, aparecerá una pantalla con huecos que corresponderán a letras de una palabra del conjunto seleccionado y una parrilla de letras aleatorias entre las que se encontrarán las pertenecientes a la palabra seleccionada.

Tal y como podemos observar en la Figura C.3 al pulsar sobre uno de los botones de la parrilla, este desaparecerá de ese lugar y se mostrará en el primer hueco vacío de la palabra preguntada. Si por el contrario, el usuario pulsara en una de las letras de la palabra preguntada, esta letra volvería a su lugar correspondiente en la parrilla y volverá a aparecer un hueco en esa posición.

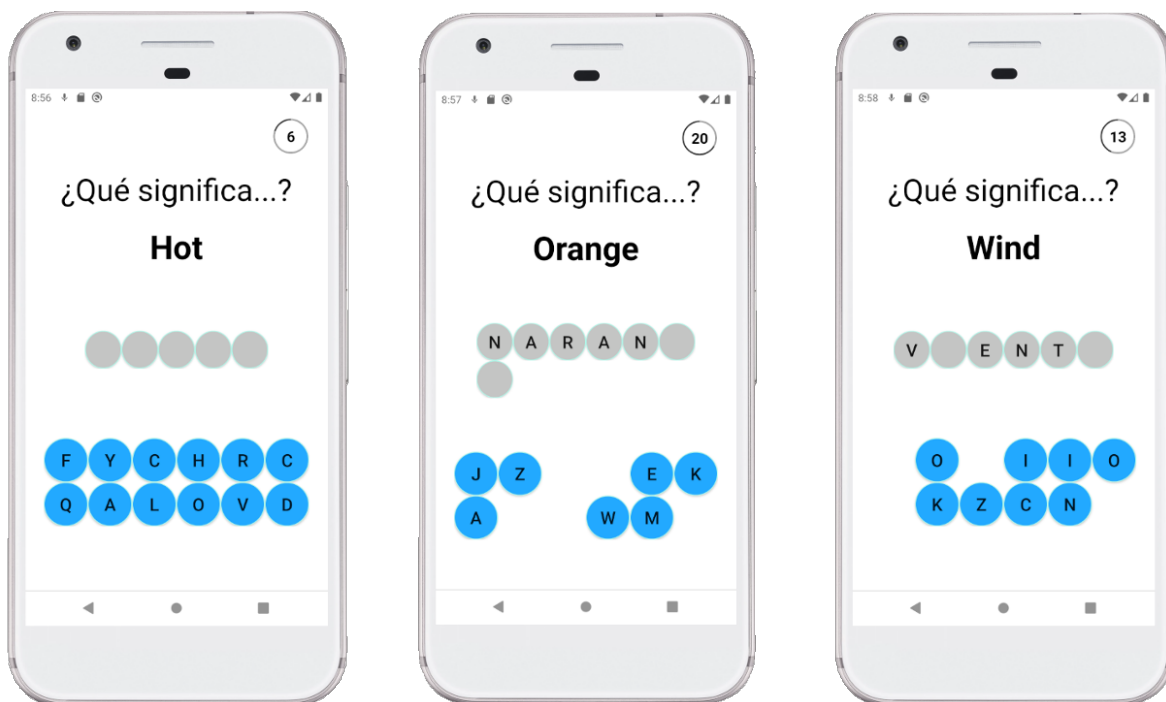


Figura C.3: Flujo de uso de la práctica Desordenado. (1)

Una vez que el usuario haya completado todos los huecos de la palabra preguntada, se comprobará si las letras introducidas son las correctas. Si el usuario no ha respondido correc-

tamente se reproducirá el mismo sonido de error de las demás prácticas, el fondo de las letras seleccionadas se volverá rojo y tras unas décimas de segundo, las letras volverán a sus lugares correspondientes en la parrilla. Sin embargo, si el usuario ha respondido correctamente, se reproducirá el sonido de acierto correspondiente, el fondo de las letras seleccionadas se tornará verde y tras unas décimas de segundo, aparecerán los huecos correspondientes a otra palabra seleccionada y otras letras distintas aparecerán en la parrilla. Este funcionamiento lo podemos visualizar en la Figura C.4.

Cabe destacar, que el usuario dispone de una cuenta atrás que disminuirá cada segundo. Sin embargo, si el usuario acierta una palabra, la cuenta atrás aumentará otros 10 segundos más. Una vez que la cuenta atrás haya llegado a 0 o el usuario haya respondido todas las palabras seleccionadas, aparecerá una pantalla final exactamente igual a la mostrada en la práctica *Test*.



Figura C.4: Flujo de uso de la práctica Desordenado. (2)

### C.1.4. Wordle

Al contrario que en el resto de prácticas, en esta no será necesario seleccionar las colecciones que el usuario quiera incluir a la hora de jugar, ya que esta práctica tiene en cuenta todas las palabras añadidas al idioma que el usuario está estudiando.

Cuando el usuario quiere comenzar a jugar por primera vez a este juego, apareceren un conjunto de huecos correspondientes a una palabra aleatoria seleccionada. El usuario deberá escribir la palabra que crea que es la que se está preguntando. Una vez haya confirmado su intento, se mostrarán las letras en los huecos de la palabra preguntada y se comparará letra por letra con la solución.

Si una letra no se encuentra en ninguna posición de la solución, el fondo de esa casilla se volverá rojo. Si una letra se encuentra en alguna posición de la solución pero no en la misma posición que en el intento que ha introducido el usuario, el fondo de esa casilla se volverá naranja. Si una letra se encuentra en la solución y además está en la misma posición que en el intento que ha introducido el usuario, el fondo de esa casilla se volverá verde. Y, por último, si una letra se encuentra más de una vez en la solución y además la posición en la que se encuentra en la solución y en el intento introducido por el usuario coincide, el fondo de esa casilla se volverá morado. Este mecanismo lo podemos visualizar en la Figura C.5.

Si el usuario no ha acertado la palabra, además de los cambios de color mencionados anteriormente, aparecerá otra fila de casillas indicando al usuario que tiene otro intento. Si, por el contrario, el usuario ha acertado la palabra, se mostrará un mensaje indicativo en el lugar donde se encontraba el botón de confirmación de intento anteriormente.

En esta práctica solo se permiten 5 intentos al día. Por lo que si el usuario deja de jugar y vuelve en el mismo día, se cargarán los intentos que ya haya realizado, y si contesta 5 veces en un mismo día, se mostrará un mensaje informativo en la misma ubicación que el botón de confirmación.

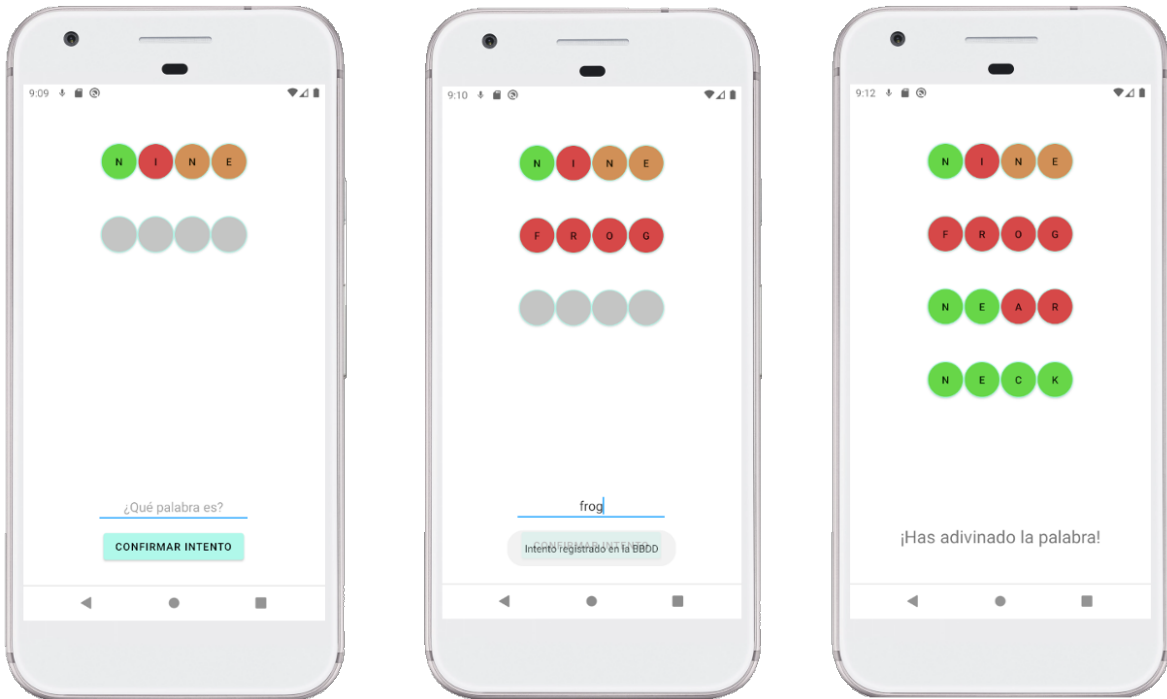


Figura C.5: Flujo de uso de la práctica Wordle.

## C.2. Demo de la aplicación

Con la finalidad de mostrar las funcionalidades de la aplicación se ha realizado un vídeo demostrativo. Como para el funcionamiento de las prácticas se requiere de al menos 10 palabras añadidas se ha optado por logearse con un usuario ya creado para ahorrar el trabajo de introducir las palabras. Este vídeo se puede encontrar en Youtube accediendo desde el siguiente enlace: <https://www.youtube.com/watch?v=LhT7k5wu9kM>.