

Ingeniería de Sistemas Audiovisuales
2018-2019

Trabajo Fin de Grado

“Desarrollo de una aplicación móvil
para la ayuda a personas con
discapacidad intelectual leve”

Daniel Ruiz García

Tutor

David Griol Barres

Leganés, junio 2019



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial**
– Sin Obra Derivada

RESUMEN

Actualmente existen fundaciones las cuales poseen programas de educación especial para personas con discapacidad intelectual donde se hace uso de las nuevas tecnologías como son los dispositivos móviles.

En este Trabajo Fin de Grado se ha desarrollado una aplicación móvil para dispositivos móviles con sistema operativo Android en la que el principal objetivo es favorecer la estimulación cognitiva de las personas con discapacidad intelectual leve. A través de dicha aplicación, los usuarios podrán practicar diversos aspectos del área lingüística, matemática, cultural y de autogestión. Además, se ha procurado que la aplicación sea lo más sencilla e intuitiva de tal manera que los usuarios puedan hacer uso de ella cuando no se encuentren con ningún responsable.

En este documento se recoge el exhaustivo análisis realizado de la discapacidad intelectual además del sistema operativo utilizado. Por otra parte, se realiza un informe detallando las principales características del entorno de desarrollo utilizado, Android Studio.

Para llevar a cabo esta aplicación se han utilizado diversas tecnologías de las cuales también se ha realizado un análisis. Se ha hecho uso del servicio de análisis de sentimiento de MeaningCloud, el motor de gestión de datos de SQLite, las bases de datos en tiempo real y el sistema de autenticación de Firebase y un sintetizador de texto a voz.

En este documento también se muestran capturas de pantalla de la aplicación con la explicación de su funcionamiento.

Además de haber realizado pruebas de la aplicación a nivel local, se ha tenido la oportunidad de realizar una visita a la clase de un plan orientado a personas con discapacidad intelectual de una fundación donde se ha podido llevar a cabo una sesión para conocer las opiniones acerca de la aplicación desarrollada y saber qué aspectos mejorar. Al final de la sesión se pasó una encuesta a los alumnos, cuyos resultados se encuentran en este documento, y los responsables del centro realizaron una carta a modo de feedback que ha sido adjuntada en el Anexo C.

Por último, en este documento también se encuentra la planificación y el presupuesto estimado del proyecto, el marco legal, las conclusiones extraídas por parte del autor y las futuras líneas de trabajo de la aplicación.

Palabras clave: discapacidad intelectual, autonomía, dispositivo móvil, Android, Android Studio, base de datos en tiempo real, sistema de autenticación, SQLite, análisis de sentimiento de MeaningCloud.

*“Lo que hacemos en la vida
tiene su eco en la eternidad”*

Gladiator

Índice de contenidos

CAPÍTULO 1. Introducción	1
1.1 Motivación.....	1
1.2 Objetivos	4
1.3 Recursos utilizados	6
1.3.1 Recursos hardware.....	6
1.3.2 Recursos software.....	6
1.4 Fases de desarrollo.....	7
1.4.1 Planificación.....	7
1.4.2 Ejecución.....	8
1.4.3 Pruebas	8
1.4.4 Cierre de proyecto.....	9
1.5 Estructura de la memoria	9
CAPÍTULO 2. Estado del arte	11
2.1 Sistema operativo Android.....	11
2.1.1 Origen	13
2.1.2 Versiones Android	14
2.1.3 Arquitectura de Android	16
2.1.4 APK	19
2.1.5 Manifiesto	20
2.1.6 Gradle.....	20
2.2 Android Studio	21
2.2.1 Interfaz de Android Studio.....	23
2.2.2 Estructura del proyecto	24
2.2.3 Entorno Java.....	25
2.2.4 SDK Android.....	26
2.2.5 Emulador	30
2.3 SQLite	30
2.3.1 Diferencias con otros gestores de bases de datos	31
2.3.2 Elementos esenciales	32
2.3.3 Inicialización de la base de datos SQLite mediante ficheros .txt	33

2.4	MeaningCloud	35
2.4.1	Ventajas de los analizadores de texto automáticos.....	35
2.4.2	APIs disponibles en MeaningCloud	36
2.4.3	Ejemplos de uso de análisis de textos automático	36
2.5	Firebase	37
2.5.1	Servicios que ofrece Firebase	37
2.5.2	Agregar Firebase a proyecto Android:.....	40
2.5.3	Sintetizador TTS (<i>Text To Speech</i>).....	42
2.5.4	Módulos de síntesis de texto a voz	43
2.5.5	TTS en Android	44
2.6	Discapacidad intelectual.....	47
2.6.1	Diagnóstico.....	48
2.6.2	Causas	50
2.6.3	Cuadro clínico asociado a la discapacidad intelectual.....	51
2.6.4	Intensidad de los apoyos	53
2.7	Análisis del mercado	54
CAPÍTULO 3. Descripción general del sistema.....		56
3.1	Descripción de la aplicación.....	56
3.2	Herramientas del sistema	57
3.2.1	Versión del SDK de compilación	57
3.2.2	Instalación de Android Studio:.....	57
3.2.3	Instalación de herramientas en SDK Manager	62
3.2.4	Configuración AVD vs. Dispositivos móviles reales:	64
3.3	Sistema de ficheros	65
3.4	Principales componentes Android utilizados:	69
3.4.1	Activity o actividad.....	69
3.4.2	Fragmento	71
3.4.3	Intents.....	72
3.4.4	Clase Application.....	72
3.5	Interfaces	73
3.5.1	Layouts.....	73
3.5.2	View o vistas	73

3.5.3	Shapes	74
3.6	Firestore. Autenticación de usuarios	75
3.7	Almacenamiento de información.....	78
3.7.1	Base de datos SQLite.....	79
3.7.2	SharedPreferences.....	80
3.7.3	Firestore. Realtime Database o base de datos en tiempo real:	81
3.8	MeaningCloud	88
3.9	Esquema resumen del sistema	89
CAPÍTULO 4. Descripción detallada del sistema.....		91
4.1	Origen del nombre del sistema	91
4.2	Inicio de la aplicación	91
4.3	Autenticación del usuario.....	92
4.4	Menú principal.....	93
4.5	Diver Lengua	95
4.5.1	Diver Frases.....	96
4.5.2	Diver Textos.....	96
4.5.3	Diver Memoria	97
4.6	Diver Mates	98
4.6.1	Diver operaciones	99
4.6.2	Diver Fracciones 1	100
4.6.3	Diver Fracciones 2	101
4.6.4	Diver Fracciones 3	102
4.6.5	Diver Euro 1	103
4.6.6	Diver Euro 2	104
4.7	Diver Trivial	105
4.8	Diver English.....	106
4.9	Autogestión	107
4.9.1	Diver Horas	108
4.9.2	Diver Ayuda	111
4.9.3	Diver Acciones	112
4.10	Sistema de puntos	113
CAPÍTULO 5. Evaluación de la aplicación		114

CAPÍTULO 6. Gestión del proyecto	121
6.1 Gestión del tiempo	121
6.2 Presupuesto estimado.....	123
6.3 Marco regulador	125
6.3.1 Políticas de desarrolladores de Google Play	125
6.3.2 Legislación	126
CAPÍTULO 7. Conclusiones y trabajos futuros	129
7.1 Conclusiones.....	129
7.2 Líneas de trabajo futuro	131
BIBLIOGRAFÍA	133
ANEXO A. Carta de la Fundación Prodis	139
ANEXO B. Glosario	140
ANEXO C. Summary.....	143
C.1 Principal elements of the application.....	143
C.2 Application	148
C.3 Evaluation of the App	149
C.4 Budget	150
C.5 Conclusion.....	150

Índice de figuras

Figura 1.1. Gráfico de personas con grado reconocido igual o superior al 33% [2]	2
Figura 1.2. Deficiencias concurridas entre 0-18 años [2].	2
Figura 1.3. Diagrama EDT	7
Figura 2.1. Evolución de la cuota de mercado de Android (2009-2018).....	12
Figura 2.2. Evolución de ventas de smartphones según su SO (2010-2018).....	12
Figura 2.3. Logo de Android	13
Figura 2.4. HTC Dream	14
Figura 2.5. Gráfico de distribución de versiones de Android	16
Figura 2.6. Arquitectura de Android [17]	17
Figura 2.7. Sistema de compilación Gradle	21
Figura 2.8. Interfaz de Android Studio	24
Figura 2.9. Estructura 'Android' (izquierda) y 'Project'(derecha) del proyecto desarrollado	25
Figura 2.10. Descarga de JDK 8.....	26
Figura 2.11. Android SDK Manager. SDK Platforms	27
Figura 2.12. Android SDK Manager. SDK Tools	28
Figura 2.13. Android SDK Manager. SDK Update Sites	29
Figura 2.14. Virtual Device ConFIGuration.....	30
Figura 2.15. Proceso de inserción de datos por medio de tabla Excel.....	35
Figura 2.16. Abrir asistente de Firebase.....	40
Figura 2.17. Asistente de Firebase.....	41
Figura 2.18. Módulos de síntesis de texto a voz.....	43
Figura 3.1. Estructura de la aplicación desarrollada	57
Figura 3.2. Descarga del archivo ejecutable de Android Studio.	58
Figura 3.3. Términos y condiciones previos a la descarga de Android Studio.	58
Figura 3.4. Asistente de instalación	59
Figura 3.5. Descarga de componentes	60
Figura 3.6. Localización de Android Studio y el SDK de Android.....	61
Figura 3.7. Instalación de Android Studio.	61
Figura 3.8. Menú de inicio de Android Studio	62

Figura 3.9. Descarga de los paquetes correspondientes a Android API 27 (Oreo 8.1).	63
Figura 3.10. SDK Tools.	63
Figura 3.11. Crear dispositivo móvil virtual en AVD Manager.	65
Figura 3.12. Sistema de ficheros del proyecto	66
Figura 3.13. Directorio <i>Manifests</i>	66
Figura 3.14. Directorio <i>java</i>	67
Figura 3.15. Directorio <i>res</i> .	68
Figura 3.16. Directorio <i>Gradle Scripts</i>	68
Figura 3.17. Pila de actividades	69
Figura 3.18. Ciclo de vida de una actividad.	70
Figura 3.19. Dos fragmentos componen una actividad.	71
Figura 3.20. Dos actividades distintas cada una con un fragmento distinto.	71
Figura 3.21. Intent-filter de la actividad ‘inicio’	72
Figura 3.22. Shape de respuestas tipo test.	74
Figura 3.23. Menú de selección de proyecto Firebase	75
Figura 3.24. Menú de selección de servicio de Firebase (En rojo todos los servicios que ofrece Firebase. En verde el servicio de Autenticación en el que se basa esta sección)	76
Figura 3.25. Pantalla principal del servicio de Autenticación de Firebase	76
Figura 3.26. Listado con todas las posibilidades de configuración del sistema de autenticación.	77
Figura 3.27. Usuarios registrados en la aplicación desarrollada	78
Figura 3.28. Inicialización de los valores de la tabla correspondiente de la base datos SQLite a través del editor de código	79
Figura 3.29. Fichero <i>.txt</i> que inicializa los valores de la tabla correspondiente de la base de datos SQLite	80
Figura 3.30. Procedimiento que ofrece Firebase para obtener el nombre del actual usuario	80
Figura 3.31. Pantalla tras acceder a la sección ‘Database’	82
Figura 3.32. Apartado de Realtime Database en la sección ‘Database’	82
Figura 3.33. Ventana de configuración de las reglas de seguridad de la base de datos en tiempo real de Firebase	83
Figura 3.34. Base de datos de Firebase vacía	83
Figura 3.35. Estructura de base de datos de Firebase	84

Figura 3.36. Esquema de los procesos llevados a cabo en la pantalla de registro de la aplicación	85
Figura 3.37. Proceso para obtener puntos de la base de datos en tiempo real de Firebase	87
Figura 3.38. Proceso seguido para actualizar una puntuación en la base de datos en tiempo real de Firebase	87
Figura 3.39. Realtime Database con datos actualizados	88
Figura 3.40. Esquema resumen de la aplicación	90
Figura 4.1. Inicio de DiverApp.....	91
Figura 4.2. Sistema de autenticación de usuarios de la aplicación desarrollada.	92
Figura 4.3. Menú principal de la aplicación desarrollada.	93
Figura 4.4. Cuadro de diálogo con el personaje de la aplicación desarrollada.	94
Figura 4.5. Cuadro de diálogo con el personaje de la aplicación desarrollada.	95
Figura 4.6. Lista de juegos de ‘Diver Lengua’	95
Figura 4.7. Juego ‘Diver Frases’	96
Figura 4.8. Juego ‘Diver Textos’	97
Figura 4.9. Juego ‘Diver Memoria’.	98
Figura 4.10. Lista de juegos de ‘Diver Mates’	99
Figura 4.11. Juego ‘Diver Mates’.	100
Figura 4.12. Juego ‘Diver Fracciones 1’.	101
Figura 4.13. Juego ‘Diver Fracciones 2’.	102
Figura 4.14. Juego ‘Diver Fracciones 3’	103
Figura 4.15. Juego ‘Diver Euro 1’.	104
Figura 4.16. Juego ‘Diver Euro 2’	105
Figura 4.17. Juego ‘Diver Trivial’.	106
Figura 4.18. Juego ‘Diver English’	107
Figura 4.19. Menú de actividades de autogestión.....	108
Figura 4.20. Menú de ‘Diver Horas’	108
Figura 4.21. Juego ‘Diver Horas 1’.	109
Figura 4.22. Juego ‘Diver Horas 2’.	110
Figura 4.23. Juego ‘Diver Ayuda’.	111
Figura 4.24. Juego ‘Diver Ayuda’.	112
Figura 4.25. Juego ‘Diver Acciones’.	112

Figura 5.1. Aplicación desarrollada en la pizarra digital	114
Figura 5.2. Alumno utilizando la aplicación	115
Figura 5.3. Formulario Google	115
Figura 5.4. Resultados para la pregunta ¿Te ha gustado el diseño de la aplicación? ...	116
Figura 5.5. Resultados para la pregunta ¿Has entendido lo que se pedía en cada juego?	117
Figura 5.6. Resultados para la pregunta ¿Te han resultado fáciles los juegos?.....	117
Figura 5.7. Resultados para la pregunta ¿Has necesitado la ayuda de algún profesor para utilizar la aplicación?	118
Figura 5.8. Resultados para la pregunta ¿Crees que podrías utilizar la aplicación tú solo sin la ayuda de un profesor?	118
Figura 5.9. Resultados para la pregunta ¿Crees que podrías utilizar la aplicación en casa?	119
Figura 5.10. Secciones de la aplicación preferidas por los alumnos	119
Figura 6.1. Diagrama de Gantt con la planificación temporal del proyecto	121

Índice de tablas

Tabla 2.1. Versiones de Android	15
Tabla 2.2. Servicios ofrecidos por Firebase dependiendo del lenguaje	40
Tabla 2.3. Clase TextToSpeech. Clases anidadas.....	44
Tabla 2.4. Clase TextToSpeech. Constantes	46
Tabla 2.5. Clase TextToSpeech. Métodos públicos sin sus parámetros de entrada	47
Tabla 2.6. Factores que influyen en la discapacidad intelectual	51
Tabla 2.7. Principales trastornos asociados a la discapacidad intelectual.....	53
Tabla 6.1. Fechas estimadas de inicio y fin del proyecto.....	122
Tabla 6.2. Costes directos. Costes de material	124
Tabla 6.3. Costes directos. Costes de personal.....	124
Tabla 6.4. Costes indirectos	124

CAPÍTULO 1. Introducción

En este primer capítulo del Trabajo Fin de Grado se procede a la exposición de la motivación para comenzar el proyecto en el que se desarrolla una aplicación móvil Android para la ayuda a personas con discapacidad intelectual leve. Además, se plasman los objetivos principales de la aplicación, los recursos utilizados durante la ejecución del proyecto, se definen las fases de trabajo y se muestra y explica la estructura que tiene el presente documento.

1.1 Motivación

Según el Manual diagnóstico y estadístico de los trastornos mentales [1] (DSM V) publicado por la Asociación Estadounidense de Psiquiatría (en inglés, APA) en 2014, la discapacidad intelectual es un trastorno que incluye limitaciones del funcionamiento intelectual y, también, en el comportamiento adaptativo en los dominios conceptual, social y práctico. Las características más comunes de las personas con discapacidad intelectual son un coeficiente intelectual (C.I.) inferior a 70-75, la afectación en ciertas habilidades de autonomía, y que se da entre los 0 y los 18 años.

Tras varios años de evolución en el concepto de este tipo de discapacidad, en 2005 nace el concepto de discapacidad intelectual o diversidad funcional cuyo objetivo es hacer una clasificación en función de la intensidad de los apoyos recibidos a pesar de que se adopte una clasificación por funcionamiento intelectual [46].

Actualmente, según los datos ofrecidos por el IMSERSO, en 2016 había en España un total de 277.472 personas cuyo primer diagnóstico ha sido la discapacidad intelectual con un grado igual o superior al 33%, lo que supone el 8,34% de la población con discapacidad. Después de las deficiencias osteoarticulares, neuromusculares, crónicas y mentales, ésta es la más común. En la Figura 1.1. se muestra en forma de gráfica una comparativa del tipo de discapacidad que poseen las personas con una discapacidad igual o superior al 30% [2].





Figura 1.1. Gráfico de personas con grado reconocido igual o superior al 33% [2]

Por otro lado, dicha discapacidad es la más común de todas en el grupo de edad de los 0 a los 34 años. La discapacidad intelectual se suele desarrollar entre los 0 y los 18 años [1]. La Figura 1.2 muestra en forma de gráfico circular una comparativa de las primeras deficiencias concurridas en las personas de edad entre los 0 y los 18 años.



Figura 1.2. Deficiencias concurridas entre 0-18 años [2].

La organización Plena inclusión [3], la cual se dedica a la inclusión de personas con discapacidad intelectual en la sociedad, afirma que aproximadamente un 1% de la población española posee algún tipo de discapacidad intelectual.

Según el documento redactado por Guevara Cuesta sobre la aproximación conceptual a la discapacidad intelectual [46], ésta se basa en un modelo multidimensional, el cual se fundamenta en las siguientes 5 premisas:

- El funcionamiento y comportamiento del individuo ha de ser medido en ambientes donde personas de edades similares juegan, trabajan e interactúan.
- Para llevar a cabo una evaluación correcta del individuo en cuestión se han de tener en cuenta aspectos como son la cultura y las costumbres individuales.
- Las personas con discapacidad intelectual tienen tantos talentos como limitaciones.
- Identificar las limitaciones únicamente es el paso previo para definir un plan de ayuda a la persona con discapacidad intelectual para que pueda mejorar su funcionamiento.
- Si las ayudas sobre el individuo persisten, se obtendrá como resultado una importante mejora funcional.

Por otro lado, Guevara Cuesta plasma cuáles son los principales aspectos en los cuales son ayudadas las personas con discapacidad intelectual:

- Las habilidades intelectuales: planificación, solución de problemas, comprensión de ideas complejas, etc.
- Participación, interacción y roles sociales.
- Salud.
- Contexto: condiciones interrelacionadas en las que el individuo convive de forma diaria.
- Conducta adaptativa:
 - Habilidades conceptuales: lenguaje comprensivo y expresivo, leer y escribir, etc.
 - Habilidades sociales: responsabilidad, relaciones interpersonales y autoestima.
 - Habilidades prácticas: actividades que se realizan de forma diaria (comer, vestirse, poner la mesa, utilizar el teléfono móvil, hacer la comida, manejar el dinero, ...).
 - Habilidades ocupacionales/laborales.

En escuelas, fundaciones y organizaciones que dan respuesta a las necesidades de las personas con discapacidad intelectual, se trabajan todos estos aspectos mediante programas específicos. Algunos de ellos, con el fin de lograr la adaptación a las nuevas tecnologías, utilizan dispositivos móviles y ordenadores.



María Ester Baños-García, Fernando Lezcano-Barbero, Esteban García-Maté y Raquel Casado-Muñoz en [84], redactan el exhaustivo análisis que han realizado con respecto a las áreas funcionales que las aplicaciones hoy en día trabajan con el objetivo de mejorar los apoyos sobre las personas con discapacidad intelectual. De las 56 aplicaciones localizadas (ya sean en inglés o en español) han podido sacar en claro que, actualmente, existe y se está desarrollando de forma progresiva una brecha tecnológica en la cual los programadores informáticos cuentan con una falta de conocimiento grande acerca de las distintas áreas de apoyo sobre las personas con discapacidad intelectual además de que algunas de las áreas tengan una cobertura exageradamente grande.

1.2 Objetivos

Actualmente, las nuevas tecnologías son utilizadas por toda la sociedad, incluso en la educación son incluidas en el proceso de enseñanza-aprendizaje (teleeducación).

Se ha llevado a cabo un estudio de las aplicaciones móviles existentes en Play Store sobre dicho tema y los resultados obtenidos han sido pobres. Existen muy pocas aplicaciones destinadas al asunto tratado, las cuales tienen un funcionamiento, en ocasiones, incluso malo y existe una ausencia de aplicaciones en castellano.

Por ello, se procede al desarrollo de una aplicación móvil orientada a personas con discapacidad intelectual que pueda ser utilizada por asociaciones, organizaciones o escuelas cuyo fin sea el desarrollo de las destrezas adaptativas de la persona a la vez que la adaptación a las nuevas formas de enseñanza. Finalmente, se concluirá el proyecto con unas pruebas de la aplicación en la Fundación Prodis donde se realizará un análisis de los resultados para sus posteriores mejoras.

El principal objetivo del desarrollo de esta aplicación Android es proporcionar una herramienta a organizaciones y asociaciones que tienen planes de ayuda para personas con discapacidad intelectual con la que se refuercen aspectos de la enseñanza y la educación, los cuales plasma Isabel García Alonso en [4], como son la toma de decisiones, estrategias de resolución de problemas, lectura, escritura, matemáticas básicas, fluidez con el sistema monetario y las habilidades de autodeterminación.

La aplicación debe consistir en una serie de actividades académicas funcionales que le sirvan a la persona con discapacidad intelectual leve para ejercitar su autonomía, es decir, fomentar que pueda desenvolverse con mayor facilidad según su criterio, con independencia.

Con el fin de favorecer el uso del lenguaje de las personas con discapacidad intelectual, se proponen actividades de comprensión y expresión de este:

- Comprensión lectora: asimilar un breve texto para la posterior evaluación mediante una serie de preguntas tipo test.
- Comprensión de ideas: retener conceptos tras una secuencia de imágenes para la selección de palabras relacionadas.



- Expresión escrita: construir frases y oraciones mediante la ordenación de palabras.

Por otro lado, uno de los objetivos es el entrenamiento en el cálculo de operaciones matemáticas básicas útiles para la vida diaria de las personas con discapacidad intelectual:

- Realizar sumas y restas básicas con números enteros.
- Realizar operaciones con fracciones.
- Entrenamiento en el manejo del sistema monetario.

Se quiere fomentar el aprendizaje del vocabulario básico de la lengua inglesa:

- Aprender palabras básicas del inglés mediante actividades lúdicas.

Además, debe existir un apartado en el que se trate de impulsar el conocimiento cultural del usuario:

- Responder cuestiones mediante un juego de preguntas tipo test de temas como son geografía, deportes, arte, historia, ciencia y literatura.

Finalmente, uno de los principales objetivos del proyecto es mejorar la autogestión de las personas con discapacidad intelectual:

- Resolución de problemas: dar respuesta a determinados problemas que pueden surgir en la vida diaria.
- Planificación: aprender cuales son los pasos que ha de seguir una persona para lograr un objetivo mediante la segmentación de actividades.
- Orientación del espacio temporal: reconocer las representaciones de las horas tanto en relojes analógicos como digitales.

Además, se han establecido una serie de objetivos orientados al correcto desarrollo de la aplicación en el sistema operativo de Android:

- Buscar y analizar las principales actividades necesarias para fomentar el aprendizaje de determinados aspectos claves en el área de la enseñanza y aprendizaje de personas con discapacidad intelectual.
- Afianzar y aumentar los conocimientos adquiridos en asignatura de Aplicaciones Móviles del grado sobre sistema operativo Android.
- La aplicación debe poseer una apariencia sencilla con botones lo más intuitivos posibles que permitan al usuario una correcta navegación por sus menús.



- Sencillez a la hora de introducir nuevas cuestiones en la aplicación mediante hojas de Excel. De esta manera, por ejemplo, en el caso de desarrollar una aplicación privada para una fundación teniendo como base este proyecto, un responsable puede actualizar de forma fácil las preguntas que aparecerán en la aplicación.
- Averiguar la forma de introducir en la aplicación un sistema de registro de usuarios.
- Introducir un módulo de autenticación de usuarios.
- Investigar y estudiar almacenamiento y obtención de datos en una base de datos en tiempo real.
- Almacenar en una base de datos en tiempo real puntos obtenidos de cada usuario de los juegos de la aplicación.
- Obtener la opinión de cada usuario sobre los juegos de la aplicación de una forma interactiva en la que el sujeto pueda continuar practicando la expresión escrita. Dicha opinión debe ser almacenada en una base de datos en tiempo real.
- Utilizar el servicio de análisis de sentimientos de MeaningCloud [33].
- Introducir un sistema de interacción por voz que pueda ser activado y desactivado.
- Realizar pruebas en la Fundación Prodis[5], tener presentes los aspectos a mejorar y llevar a cabo un análisis de cuan efectiva es la aplicación.

1.3 Recursos utilizados

Se procede a distinguir los recursos utilizados para el desarrollo de la aplicación:

1.3.1 Recursos hardware

- Ordenador portátil ASUS Transformer Book Flip (I5)
- iMAC 9.1
- Cable USB
- Teléfonos móviles

1.3.2 Recursos software

- Android Studio (versión 3.0.1)
- Software Development Kit (SDK): contiene todas las herramientas necesarias para crear una aplicación Android en Windows, Mac OS y Linux.
- Software Development Kit Manager (SDK Manager): permite la actualización de las versiones de Android instaladas en SDK o instalar nuevas.
- GIMP
- Image Alpha: software utilizado para llevar a cabo una reducción del tamaño de archivos 24bit PNG a PNG8.



- Image Optim: busca la forma de optimizar el espacio que ocupan las imágenes eliminando comentarios innecesarios y buscando factores de conversión mejores.
- Windows 10.
- Microsoft Office 365 ProPlus.
- Keynote.

1.4 Fases de desarrollo

Para llevar a cabo una correcta gestión del proyecto, se han utilizado los conocimientos adquiridos en la asignatura Proyectos, normativa y política de las telecomunicaciones en la cual se podían distinguir 5 fases: inicio del proyecto, planificación, ejecución, seguimiento y control o pruebas y cierre.

La Figura 1.3 muestra un diagrama EDT (Estructura de Descomposición de Trabajo) o también conocido como WBS (Work Breakdown Structure) en el que, a partir de las 5 fases definidas anteriormente, se registran de forma jerárquica las correspondientes subactividades que hay que llevar a cabo para lograr el objetivo final.

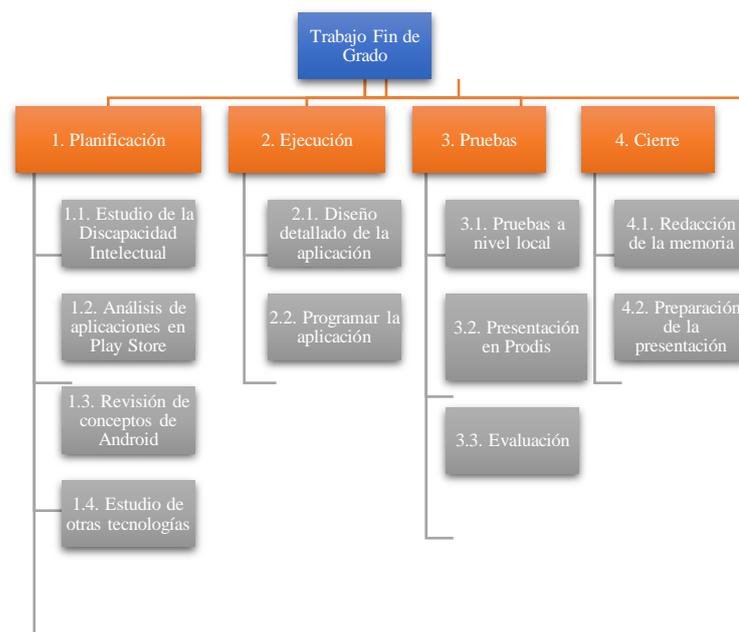


Figura 1.3. Diagrama EDT

Como se puede observar, el proyecto está dividido en los siguientes bloques:

1.4.1 Planificación

- **Estudio de la Discapacidad Intelectual (DI):** a través de artículos encontrados en internet, páginas web y consultas a profesionales de la enseñanza especial, se ha llevado a cabo un estudio de lo que es la discapacidad intelectual, cómo se produce, la clasificación por niveles y los

diferentes planes de ayuda que existen hoy en día orientados a las personas que lo poseen.

- **Análisis de aplicaciones en Play Store:** antes de comenzar un proyecto de desarrollo de una aplicación Android, es importante realizar un análisis exhaustivo de las aplicaciones existentes en el mercado. Para ello, se ha accedido a Play Store, la actual plataforma de distribución de aplicaciones Android y se ha procedido a la descarga de algunas de las aplicaciones orientadas a la discapacidad intelectual para hacer un análisis.

Este paso tiene una gran importancia ya que además de querer lograr el desarrollo de una aplicación orientada a la discapacidad intelectual leve, uno de los objetivos es crear una aplicación que sea diferente al resto de las existentes, es decir, una aplicación la cual al descargarla un usuario pueda detectar que es notablemente mejor y cubre mayores áreas que el resto.

- **Revisión de conceptos de la programación Android:** a pesar de haber realizado otra aplicación móvil Android durante el año 2018 en la asignatura de Aplicaciones Móviles, algunos de los conceptos de la plataforma Android habían sido olvidados. En [6] se han podido consultar aquellos aspectos básicos de Android que no se recordaban.
- **Estudio de otras tecnologías:** en esta fase del proyecto fin de carrera se han estudiado gestión de datos en bases de datos SQLite y en bases de datos en tiempo real (Firebase), módulos de registro y de autenticación de usuarios (Firebase) y el servicio de análisis de sentimientos de MeaningCloud.

1.4.2 Ejecución

- **Diseño detallado de la aplicación:** una vez realizados todos los estudios previos, escoger el área sobre el que se va a trabajar y tener una idea de lo que se quiere desarrollar, se procede a un primer diseño de la aplicación a la vez que se es consciente de que puedan surgir cambios. Se eligen las funcionalidades de la aplicación y se realizan bocetos de las diferentes pantallas que contendrá la aplicación.
- **Programar la aplicación:** se procede a desarrollar cada una de las funcionalidades y pantallas de la aplicación mediante la plataforma de Android Studio.

1.4.3 Pruebas

- **Pruebas a nivel local:** una vez programada la aplicación se llevan a cabo una serie de pruebas a nivel local de tal manera que, ante cualquier posible error de programación se pueda actuar de manera eficiente realizando una modificación de código.



- **Pruebas en Fundación Prodis:** tras comprobar el correcto funcionamiento de la aplicación, el día 21 de mayo de 2019 se pudo efectuar pruebas además de una presentación del proyecto en la Fundación Prodis, a la que se le agradece su colaboración. La Fundación Prodis tiene como objetivo principal mejorar la calidad de vida de las personas con discapacidad intelectual a la vez que promueve la inclusión social justa y solidaria [7].
- **Evaluación:** como parte de las pruebas en la Fundación Prodis, se ha pedido la participación en una encuesta en la cual se preguntaba por la opinión de los usuarios de la aplicación y el personal de la fundación.

1.4.4 Cierre de proyecto

- **Redacción de la memoria:** como parte final del Trabajo Fin de Grado, se ha procedido a la escritura del presente documento donde se describen con detalle cada uno de los procedimientos y conceptos que conforman la aplicación móvil Android.
- **Preparación de la presentación:** el día 4 de julio de 2019 se llevará a cabo la presentación del proyecto fin de carrera ante el tribunal.

1.5 Estructura de la memoria

- **Capítulo 1: Introducción.**

En este capítulo del documento se presentan los principales objetivos del desarrollo de la aplicación, la motivación de realizarlo, los recursos utilizados para llevarlo a cabo, una distribución jerárquica de las fases del proyecto y la estructura que tiene la memoria.

- **Capítulo 2: Estado del arte.**

En este capítulo se lleva a cabo un estudio de dos ámbitos: la discapacidad en cuestión y el tecnológico. En el área de la discapacidad intelectual se explican cuestiones como qué es, cómo clasificarla, cómo detectarla y cuáles son sus causas. Por otro lado, en el dominio tecnológico se detallan los estudios realizados de bases de datos SQLite, bases de datos en tiempo real de Firebase, módulos de registro y autenticación de usuarios, sintetizadores de texto a voz y el servicio de análisis de sentimientos que ofrece MeaningCloud.

Además, se muestra al lector del presente documento el análisis realizado sobre otras aplicaciones existentes orientadas a la ayuda de personas con discapacidad intelectual leve en Play Store.

- **Capítulo 3: Descripción general del sistema.**

En este capítulo se hace una descripción general del funcionamiento de la aplicación. En él se explicará su estructura o sistema de ficheros, las herramientas o recursos necesarios y se realizará un análisis a fondo sobre el núcleo de la aplicación.



- **Capítulo 4: Descripción detallada del sistema.**

En este capítulo se explica por medio de capturas de pantalla cada una de las actividades que componen el sistema además de presentar el nombre de la aplicación y su origen.

- **Capítulo 5: Evaluación de la aplicación.**

El 21 de mayo de 2019 se realiza una presentación y unas pruebas del sistema en la Fundación Prodis. Después de ello, se ha procedido a la distribución de una encuesta para en este capítulo realizar un análisis de los resultados.

- **Capítulo 6: Gestión del proyecto.**

En este capítulo se ha procedido al desarrollo de un diagrama de Gantt en el cual se puede visualizar de una forma sencilla el tiempo invertido para cada una de las actividades que componen las fases de este proyecto.

Por otro lado, se realiza una estimación de cuál sería el presupuesto que supondría realizar el proyecto desde cero. En el cálculo se tendrá en cuenta el coste el personal que ha participado en el proyecto, el material utilizado, los servicios contratados, etc.

Finalmente, se exponen los principales aspectos legales a tener en cuenta para desarrollar esta aplicación.

- **Capítulo 7: Conclusiones y trabajos futuros.**

En este capítulo se exponen las posibles futuras líneas de desarrollo en la además de una conclusión del Trabajo Fin de Grado.

- **ANEXO A. Carta de la Fundación Prodis.**

Se adjunta a modo de anexo la carta redactada por los profesionales de la Fundación Prodis después de la visita realizada el 21 de mayo de 2019.

- **ANEXO B. Glosario.**

Se exponen las definiciones o explicaciones de los principales términos que componen la memoria del proyecto. De esta manera, se facilita la lectura del documento.

- **ANEXO C. Bibliografía.**

En este apartado se mencionan todas y cada una de las fuentes que se han consultado tanto durante el desarrollo de la aplicación como la redacción del presente documento.



CAPÍTULO 2. Estado del arte

En este segundo capítulo de la memoria, se lleva a cabo un análisis del sistema operativo Android en el que se expone una definición, su origen las versiones existentes hoy en día y la arquitectura que posee. Asimismo, se define el APK, el manifiesto y el sistema de compilación Gradle.

También se realiza un análisis del software Android Studio; qué es, sus principales funciones, se presenta su interfaz y la estructura del proyecto. De igual forma, se define el entorno de desarrollo Java, el SDK y la herramienta de emulador.

Por otro lado, se muestran los aspectos principales de los módulos de conversión de texto a voz, los sistemas de gestión de bases de datos SQLite y el analizador de sentimiento proporcionado por MeaningCloud. También se puede observar el abanico de posibilidades que tiene una aplicación de ser conectada a una cuenta Firebase (módulo de registro y autenticación, base de datos en tiempo real).

Se realiza un estudio de la discapacidad intelectual: qué es, cómo clasificarla, el cuadro clínico asociado a este tipo de discapacidad, cómo detectarla y cuáles son sus causas.

Por último, se lleva a cabo un análisis del mercado haciendo una comparación de algunas de las aplicaciones encontradas que se orientan al asunto que se trata.

2.1 Sistema operativo Android

Android es un sistema operativo que fue desarrollado por Google con el objetivo de poder implantarlo en los dispositivos móviles con pantalla táctil [9]. Este sistema operativo está basado en Linux, otro sistema operativo el cual es un ejemplo claro de código abierto y software libre [9][11]. Código abierto ya que los códigos fuente tienen carácter público y todo el mundo puede acceder a ellos [12] y software libre porque el código fuente podrá ser utilizado para cualquier fin, incluida su redistribución [13].

Actualmente, este sistema operativo está en auge estando presente en el 90% de las ventas de dispositivos móviles, ya sean tablets, teléfonos móviles o relojes. Se han obtenido los datos del artículo escrito por García Nieto [42] para mostrar en forma de gráfica en la Figura 2.1. a través de la herramienta [43], la evolución de la cuota de mercado desde 2009 hasta 2018 de los principales sistemas operativos.



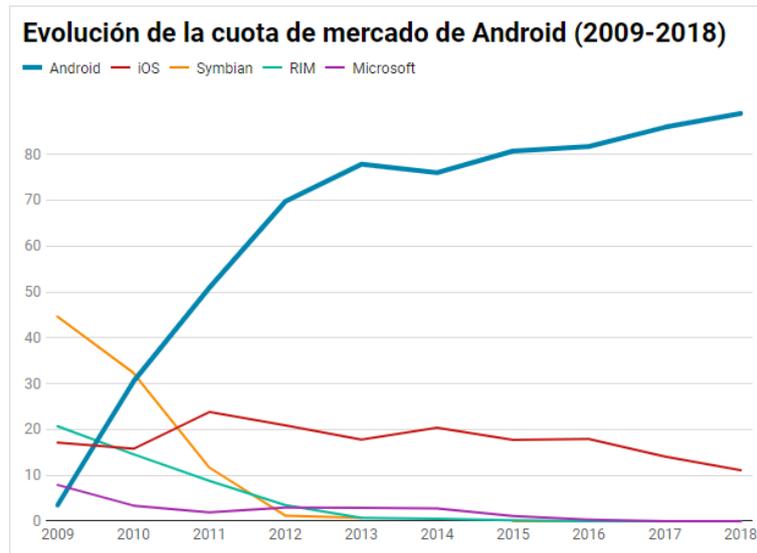


Figura 2.1. Evolución de la cuota de mercado de Android (2009-2018)

Como se puede observar en la Figura 2.2, en los inicios de Android, allá por el 2008/2009 cuando apareció el primer SDK Android, el sistema operativo Symbian era líder y el resto de los principales sistemas operativos estaban por encima de Android. Solamente le bastó un año a Android para superar a su actual principal competidor, iOS, y ser el líder del mercado. Desde entonces, el sistema operativo en cuestión ha protagonizado un crecimiento continuo hasta la actualidad, que sigue creciendo.

Y es que, de la mano de estos datos, viene la enorme evolución de Android frente al resto de competidores con respecto al número de ventas de dispositivos móviles con un sistema operativo u otro. Nuevamente, se han obtenido los datos ofrecidos en el artículo [42] para expresar en forma de gráfica en la Figura 2.2. a través de la herramienta [43] la evolución de ventas de smartphones según el sistema operativo:

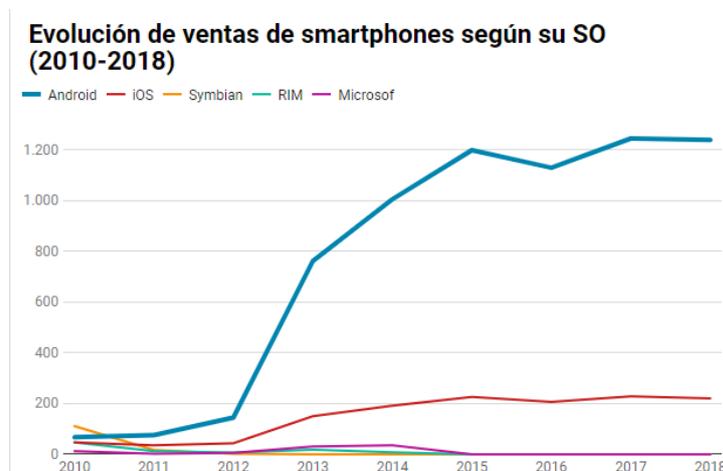


Figura 2.2. Evolución de ventas de smartphones según su SO (2010-2018)

Sin embargo, a pesar del dominio de Android en estos aspectos y además de ser líder en descargas de aplicaciones en Google Play, la cantidad de dinero que ingresa es mucho menor que el dinero que está ingresando iOS con la mitad de las descargas en App Store [44].

Es por todo ello por lo que merece la pena desarrollar aplicaciones Android. Un sistema operativo de código abierto en el que una gran comunidad de programadores hace que cada vez crezca más.

2.1.1 Origen

En octubre de 2003 Andy Rybin, Rich Miner, Nick Sears y Chris White dieron lugar en Palo Alto (California) a la empresa de Android.

En el año 2005 la multinacional Google compra Android y dos años más tarde, en 2007, se crea la *Open Handset Alliance* (OHA). Se trata de una alianza que actualmente cuenta con 84 empresas que se dedican a generar código abierto para dispositivos móviles. Por aquel entonces, en 2007, Google se encontraba al frente de la OHA junto a otros 34 miembros [8]. Android ahora era de manera oficial una *open source*, es decir desarrollaba código fuente de manera pública. Ese mismo año, el logo de la empresa fue diseñado por Irina Blok [9]. Se trataba de un androide:



Figura 2.3. Logo de Android

Un año más tarde, en 2008, fue cuando apareció la primera versión del SDK Android 1.0 y con ella, el primer teléfono móvil con Android. Se trataba del HTC Dream [6].



Figura 2.4. HTC Dream

2.1.2 Versiones Android

Como se ha explicado anteriormente, Android desde 2008 ha actualizado en muchas ocasiones las versiones de desarrollo y cada una de ellas diferente a las anteriores, siempre añadiendo nuevas funcionalidades y arreglando errores de las anteriores. Sin embargo, a medida que aparecen nuevas, el número de dispositivos móviles que no las soportan es mayor debido a que su hardware fue diseñado para funcionar de manera correcta con las versiones anteriores. Por tanto, se puede decir que Android de alguna manera está impulsando la obsolescencia de algunos dispositivos móviles [9].

Es por ello por lo que, antes de comenzar el desarrollo de la aplicación, es importante saber cuáles serán las funcionalidades que la conformarán. De esta manera, podremos escoger la versión de Android correcta en la que programar y con ello el nivel de API. Hay que escoger entre, o bien desarrollar una aplicación cuyas funcionalidades sean las más recientes que pone a nuestra disposición Android o bien que la aplicación sea accesible de forma correcta por el mayor número de dispositivos. En el desarrollo de esta aplicación móvil se ha optado por 8.1 con nivel de API 27.

A cada una de las actualizaciones de Android se le asigna un nombre de dulce y siguen un orden alfabético. A continuación, en la tabla 2.1. se muestra una tabla de todas las actualizaciones realizadas por Android hasta la fecha [6]:

Nombre	Versión	Fecha de lanzamiento	API	Distribución
Apple Pie	1.0	Septiembre 2008	1	-
Banana Bread	1.1	Febrero 2009	2	-
Cupcake	1.5	Abril 2009	3	-

Donut	1.6	Septiembre 2009	4	-
Eclair	2.0 - 2.1	Octubre 2009	5 – 7	-
Froyo	2.2 - 2.2.3	Mayo 2010	8	-
Gingerbread	2.3 -2.3.7	Diciembre 2010	9 – 10	0,3%
Honeycomb	3.0 - 3.2.6	Febrero 2011	11 – 13	-
Ice Cream Sandwich	4.0 - 4.4.4	Octubre 2011	14 – 15	0,3%
Jelly Bean	4.1 - 4.3.1	Julio 2012	16 – 18	3,2%
KitKat	4.4 - 4.4.4	Octubre 2013	19 – 20	6,9%
Lollipop	5.0 – 5.1.1	Noviembre 2014	21 – 22	14,5%
Marshmallow	6.0 – 6.0.1	Octubre 2015	23	16,9%
Nougat	7.0 – 7.1.2	Junio 2016	24 – 25	19,2%
Oreo	8.0 – 8.1	Agosto 2017	26 – 27	28,3%
Pie	9.0	Agosto 2018	28	10,4%
“Q”	10.0	Agosto 2019	29	-

Tabla 2.1. Versiones de Android

En la Figura se muestra un gráfico de la distribución estadística de cada una de las versiones Android disponibles hasta la fecha. Los datos fueron recopilados durante un periodo de 7 días hasta la fecha 26/10/2018 [10].

GRÁFICO DE DISTRIBUCIÓN DE VERSIONES ANDROID



Figura 2.5. Gráfico de distribución de versiones de Android

2.1.3 Arquitectura de Android

La arquitectura de Android se basa en un sistema de capas en el cual cada capa utiliza las inferiores y a su vez presta sus servicios a las superiores. Es decir, se trata de una pila de software donde se incluyen las aplicaciones del sistema, el Framework de las aplicaciones, las bibliotecas, un módulo de ART y el núcleo Linux. Dichas capas que se pueden visualizar en la Figura 2.3.

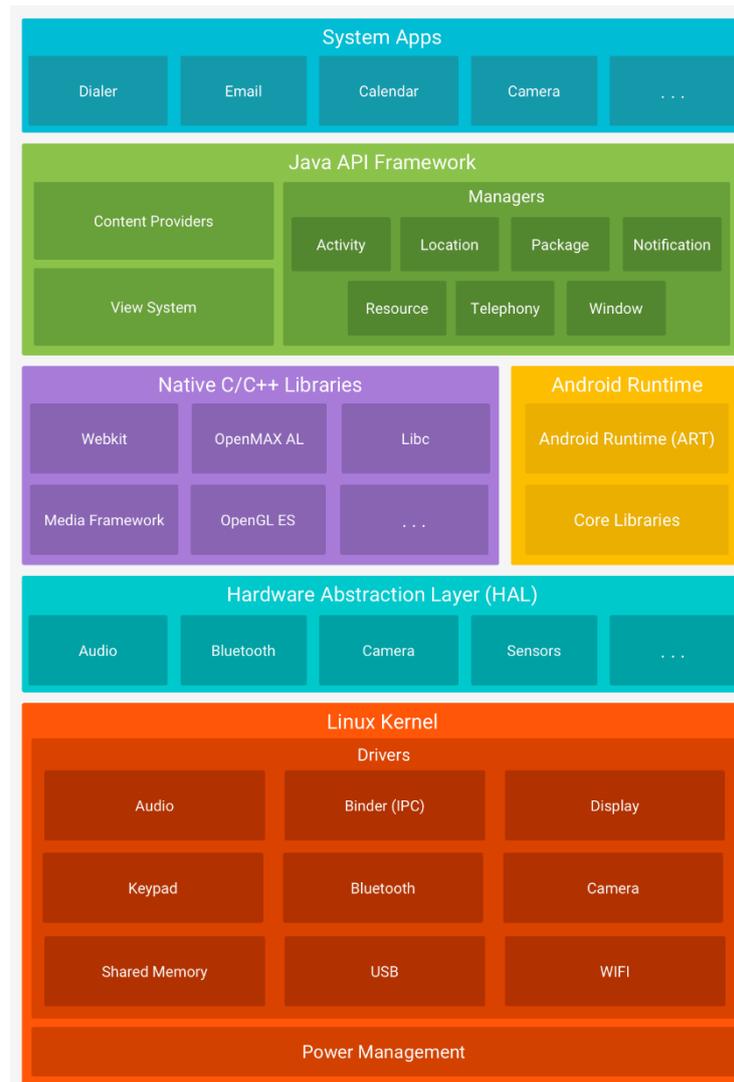


Figura 2.6. Arquitectura de Android [17]

- **Núcleo de Linux:** Android a partir de la versión 5.0 (API 21) pone en funcionamiento el modelo de tiempo de ejecución de Android Runtime, el cual se basa en Linux y se explicará más a fondo en capa de tiempo de ejecución. Además, Linux permite el uso de funciones de seguridad claves y actúa como una capa de abstracción entre el hardware y el resto de la pila del software [17].

Por otro lado, la capa del kernel de Linux posee los drivers necesarios para ser utilizado por cualquier hardware a través de una llamada. Cuando los fabricantes desean incluir un elemento nuevo de hardware, han crear de los drivers o librerías de control necesarias dentro del núcleo de Linux que está embebido en Android para que a través del código pueda ser utilizado, como por ejemplo el uso del bluetooth del dispositivo móvil [18].

- **Capa de abstracción de hardware:** es la capa que contiene las interfaces correspondientes de cada uno de los componentes hardware del dispositivo

móvil. De esta manera si el framework desea utilizar un elemento hardware, realiza una llamada y se carga el módulo con la interfaz propia del elemento en cuestión [18].

- **Tiempo de ejecución de Android:**

- Dalvik vs. ART

El sistema operativo de Android permite la programación de aplicaciones a partir de Dalvik, una variación del lenguaje de Java [14] que se basa en la compilación en tiempo de ejecución (*Just inTime*). Dalvik es la máquina virtual que posee ciertas variaciones con respecto a las máquinas virtuales de Java que utilizan algunos de los dispositivos Android. Esto se debe a que las instrucciones que la máquina virtual espera recibir, es decir el bytecode, son distintas a las de Java. Por otro lado, Dalvik es utilizado ya que se basa en registros y no en pilas, mejorando el rendimiento de los dispositivos móviles y, además, el uso de memoria es pequeño. De esta manera, lo que Android pretende al utilizar Dalvik es poder ser flexible frente a cualquier dispositivo móvil, es decir, ser lo suficientemente eficiente para dispositivos móviles de mayor o menor potencia de manera independiente [15].

En el proceso de crear una aplicación Android, en primer lugar, se genera un archivo *.class* el cual se convierte a un archivo Dalvik ejecutable *.dex* y finalmente se comprime en el archivo APK (*Android application package*) [15].

Sin embargo, desde la versión 5.0, la máquina virtual utilizada para la programación de aplicaciones Android es ART (*Android Runtime*). A diferencia de Dalvik, esta máquina virtual realiza la compilación en tiempo de instalación y no de ejecución, algo que potencia de manera grande el rendimiento de las aplicaciones [6].

Por otro lado, el sistema de gestión de memoria utilizado, *Garbage Collector*, es mejorado con ART [6]. En el sistema de trabajo de *Garbage Collector*, en primer lugar, se reservan espacios de memoria. A medida que se dejan de utilizar, se liberan para posteriormente compactar todos los espacios de memoria utilizados y dejar todos los libres juntos. Finalmente, se comprueban cuales están libres para volver a ser utilizados [16].

Por último, ART ofrece mensajes de error más detallados que Dalvik [6].

- **Bibliotecas C/C++ Nativas:** algunos de los componentes del sistema de Android hacen uso de unas bibliotecas en lenguaje C o C++ con el objetivo de ofrecer un determinado servicio [18]. Algunas de ellas son:

- SQLite: crear y gestionar bases de datos.



- Surface Manager: compone elementos de navegación de pantalla.
- MediaLibraries: proporciona códecs para el contenido multimedia que soporta Android.
- Libc: proporciona cabeceras funciones según el estándar del lenguaje C.
- FreeType: proporciona facilidades para trabajar con distintos tipos de fuente.
- **Framework:** se encuentran las principales herramientas para implementar cualquier funcionalidad de Android en el desarrollo de una aplicación. Se trata de las APIs, a las que todos los desarrolladores tienen acceso en su totalidad. Algunas de las APIs más importantes son [18]:
 - Activity Manager: gestiona el ciclo de vida de las aplicaciones.
 - Content Provider: gracias a ella aplicaciones como la agenda de contactos puede compartir su información con el resto de las aplicaciones del dispositivo.
 - Location Manager: aplicaciones como Google Maps utilizan esta API para llevar a cabo una localización y un posicionamiento del usuario.
 - Notification Manager: gestiona los mensajes que las aplicaciones envían al usuario mediante notificaciones.
 - Telephone Manager: permite a la aplicación hacer uso de las funcionalidades del teléfono.
- **Apps del sistema:** en esta capa se encuentran las aplicaciones base del dispositivo, es decir, todas ellas de las que puedes hacer uso a partir de otra distinta descargada. Por otro lado, también se encuentran las aplicaciones que el usuario ha decidido instalar. Si el desarrollador desea crear una aplicación en la cual pueda realizar llamadas, no hará falta que implemente la funcionalidad; directamente se podrá hacer uso de la aplicación base o la instalada por el usuario encargada de realizar llamadas [17].

2.1.4 APK

El archivo APK (*Android application package*) es el resultado de compilar y empaquetar una aplicación Android. Este tipo de archivo contiene [6]:

- El archivo *.dex* explicado en la sección 2.1.3.
- La carpeta de recursos utilizados durante el desarrollo de la aplicación
- Certificados
- El archivo manifiesto, del cual se da una explicación en la sección 2.1.5.



2.1.5 Manifiesto

Todas las aplicaciones deben poseer este archivo en el que se proporciona cierta información para poder ejecutar el código. En primer lugar, se ha de especificar el nombre del paquete de la aplicación. Posteriormente se han de configurar las siguientes secciones [6]:

- **Permissions:** se declaran los permisos necesarios para acceder a partes protegidas de una API y que la aplicación se ejecute de forma correcta.
- **Uses-feature:** se especifican los elementos hardware que utiliza la aplicación, como por ejemplo la cámara.
- **Supports-screens:** se especifica el tamaño de la pantalla permitido para que la aplicación se ejecute de forma correcta.
- **Application:** se definen cada uno de los componentes de la aplicación. Además, se deberá detallar:
 - Nombre a la aplicación
 - Icono
 - Cada uno de los componentes de la aplicación y las clases que implementa cada uno.

2.1.6 Gradle

Gradle es un plugin el cual es un sistema de compilación basado en JVM (*Java Virtual Machine*) [21]. Las máquinas virtuales java JVM son capaces de ejecutar código generado por el compilador de lenguaje Java [22].

Google creó este lenguaje con el fin de que los programadores no se viesen obligados a aprender un lenguaje nuevo permitiendo que cada desarrollador pueda escribir su propio fichero [21].

El proceso seguido por el sistema de compilación Gradle es el siguiente [6]:

- 1) Compilar código fuente, recursos y librerías utilizadas dando como resultado los archivos *.dex*.
- 2) Se les aplica una firma mediante keystore, es decir, se le aplica una clave criptográfica para que resulte más complicado extraerlo del dispositivo.
- 3) El conjunto de los archivos compilados con la firma conforma el APK.

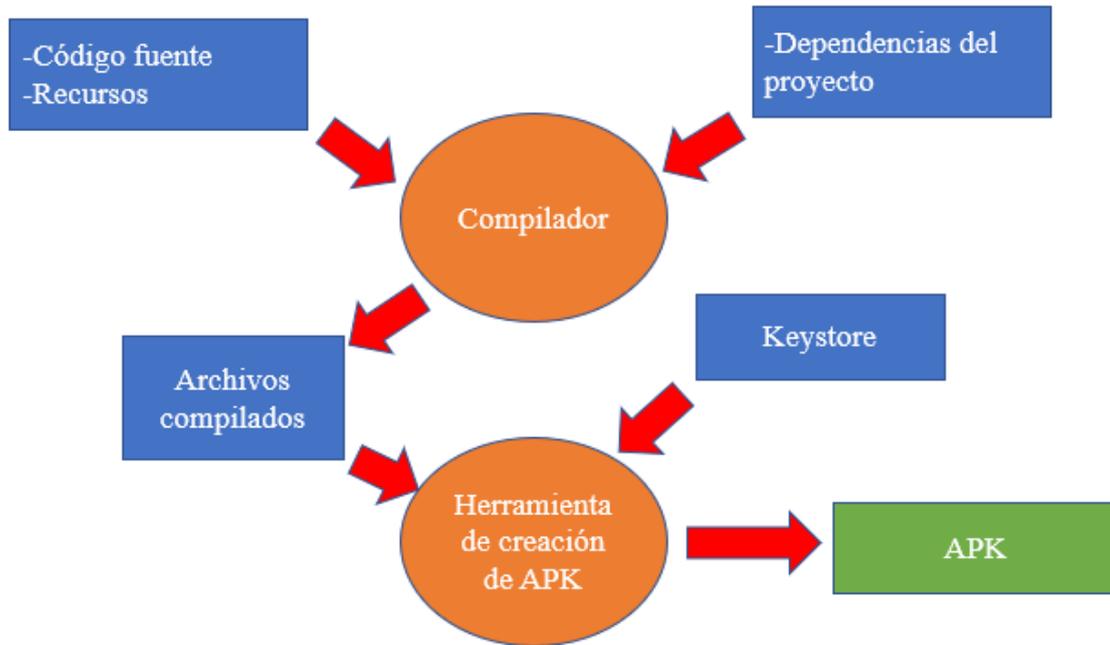


Figura 2.7. Sistema de compilación Gradle

Existen dos archivos los cuales sirven para configurar y personalizar la compilación gradle. Por un lado existe *settings.gradle* que es el archivo en el que se informa de los diferentes módulos que componen el proyecto. Por otro lado existe *build.gradle (Project:app)* que es el archivo gradle que contiene la configuración que es común a todos los módulos que componen el proyecto[6]. También existe un archivo *build.gradle (Module:app)* en el cual se define la información específica de cada módulo que compone la aplicación. En él, además, se definen las dependencias del módulo, es decir, las implementaciones que han sido necesarias para llevar a cabo el desarrollo de la aplicación.

2.2 Android Studio

Tras sustituir a Eclipse en mayo de 2013, se trata del entorno de desarrollo integrado utilizado para la creación de nuevas aplicaciones para el sistema operativo Android. Además, está basado en lo que antes se conocía como IntelliJ IDEA [6] y ahora se conoce como JetBrains, que es otro entorno de desarrollo integrado (IDE) destinado al desarrollo de programas informáticos [20].

Antes de comenzar a desarrollar una aplicación es importante saber cuál será el entorno de desarrollo. En este caso, parece estar claro que se ha escogido Android Studio. Pero ¿por qué? Eclipse es otro IDE que muchos programadores han utilizado a lo largo del tiempo para desarrollar nuevas aplicaciones Android. Con la instalación de un plugin de ADT (*Android Development Tools*) se era capaz de programar aplicaciones Android. A continuación, se procede a clasificar las principales funciones que ofrece Android Studio y que le hacen ser mejor y diferenciarse de Eclipse [26]:

- Es capaz de proporcionar las herramientas necesarias para que la escritura del código y las iteraciones sean lo más veloz posible.
 - *Instant Run*: una vez ejecutada la aplicación mediante el botón ‘*Run*’ o ‘*Debug*’, Android Studio mediante ‘*Instant Run*’ es capaz de detectar las modificaciones que se hayan realizado en el código para que no sea necesario reiniciar o compilar el APK.
 - Editor de código inteligente: Android Studio ofrece una serie de sugerencias de autocompletado al mismo tiempo que escribes. Simplemente posicionando el cursor sobre la sugerencia deseada y pulsando la tecla tabuladora, el sistema realiza la función de autocompletado.
 - Emulador rápido y multitud de funciones: el emulador que posee Android Studio carga la aplicación más rápido que un dispositivo móvil. Además, cuenta con ciertas propiedades configurables para poder hacer simulaciones virtuales en diversos dispositivos (tipo de dispositivo, resolución de pantalla y tamaño de pantalla).
- Gracias a la compilación y estructura de proyectos de Android Studio basada en Gradle, es sencillo generar un APK que se adapte al dispositivo.
 - Sistema de compilación sólido y flexible: es posible incorporar al proyecto tanto bibliotecas locales como externas, realizar variaciones en el código y aplicar configuraciones de reducción de código y firma de la aplicación.
 - Existe un sistema de control de versiones del proyecto y se puede adaptar la configuración al entorno.
 - Android Studio es capaz de compilar proyectos tanto para tablets y teléfonos móviles Android, Android Auto, Android TV y Android Wear.
- Con Android Studio el usuario puede estar seguro de que el código desarrollado es el mejor posible.
 - Cuenta con diversas plantillas de código y ejemplos de diversos aspectos de la programación Android que se pueden importar y ejecutar.
- Se puede desarrollar código que no sea en lenguaje Java ni se ejecute en el dispositivo del usuario.
 - Se pueden compilar componentes JNI, es decir, la aplicación desarrollada en Java es capaz de interactuar con otros programas que estén escritos en C o C++ [27].
 - Android Studio te permite incorporar a la aplicación sistemas de autenticación, análisis estadístico, notificaciones, bases de datos en tiempo real, ...a partir de Firebase. El software Android Studio cuenta con un tutorial donde se explica cómo conectar paso a paso con Firebase Por otro



lado, Google Cloud Platform también ofrece una variedad de módulos al usuario.

- Android Studio cuenta con una gran variedad de herramientas de interfaces gráficas de usuario GUI (*Graphical User Interface*).
 - En la edición de archivos .xml, se ofrece un editor de diseño en el cual se facilita mucho las tareas del programador. En él, simplemente se deberá arrastrar los elementos que se desean al lugar correspondiente.
 - Mediante un análisis de APK se puede conocer el tamaño de cada uno de sus componentes para tratar de reducirlo, visualizar los recursos empaquetados y realizar una comparación entre dos APKs.
 - Con *Vector Asset Studio* se pueden generar una imagen para diversas densidades a partir de iconos proporcionados por Google o imágenes PSD o SVG importadas por el usuario (diseño vectorial). Con el fin de que dispositivos móviles más antiguos puedan hacer uso de recursos vectoriales en el caso de que no admitan elementos de diseño vectoriales de Android, *Vector Asset Studio* permite a su vez genera mapas de bits en distintas densidades.
 - Gracias a *Translations editor*, el usuario puede visualizar todos los recursos utilizados en el proyecto traducidos. En él se pueden modificar o añadir traducciones.

Por todas estas funcionalidades que cuenta Android Studio, se ha decidido desarrollar la aplicación del proyecto con este entorno de trabajo. Además, actualmente, Eclipse ADT se considera un producto obsoleto ya que son una gran cantidad de herramientas las que se pierden si no se utiliza Android Studio. Es más, Google, el 26 de junio de 2015, advierte que su equipo de programadores se ha centrado única y exclusivamente en el desarrollo de aplicaciones con Android Studio y sugiere que los usuarios que aún utilizan Eclipse ADT migren sus proyectos a Android Studio [45].

2.2.1 Interfaz de Android Studio

En esta sección de la memoria se presentan las distintas partes que contiene la interfaz del software de Android Studio. A continuación, en la Figura 2.8. se puede observar la distribución que sigue el programa:



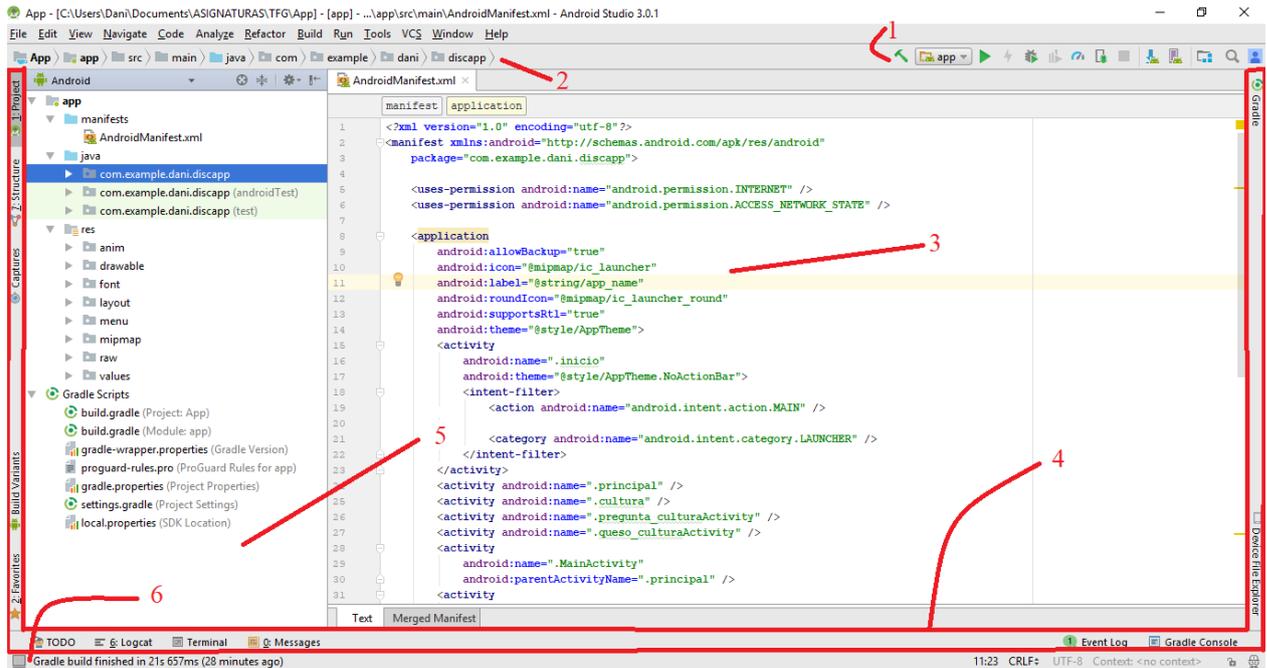


Figura 2.8. Interfaz de Android Studio

1. Barra de herramientas: es el lugar desde donde el usuario puede ejecutar el proyecto, realizar una ejecución del proyecto con *debug*, abrir el menú del software *SDK Manager*, ...
2. Barra de navegación: se puede ver el directorio del archivo abierto en Android Studio además de ser una forma más sencilla y compacta de la estructura del proyecto.
3. Ventana del editor: es espacio reservado para que el desarrollador pueda crear o modificar código.
4. Barra de la ventana de herramientas: es el lugar desde donde se pueden expandir diversos menús correspondientes cada uno a herramientas distintas como puede ser la terminal o el Logcat.
5. Ventana de herramientas: corresponde al lugar desde donde se puede observar la estructura del proyecto además de poder abrir en la ventana del editor los archivos.
6. Barra de estado: es la zona de la pantalla donde se pueden observar los mensajes que Android Studio manda al usuario sobre el estado del proyecto.

2.2.2 Estructura del proyecto

Una aplicación móvil Android está compuesta por uno o varios módulos los cuales pueden tratarse de otra aplicación distinta, de bibliotecas externas o del servicio de alojamiento web gratuito que ofrece Google, Google App Engine [23] [24]. Por otro lado, cada uno de estos módulos poseen las carpetas:



- Java: donde se encuentran subcarpetas con el nombre del paquete y además se puede encontrar los ficheros *.java* que contienen el código fuente de la aplicación en lenguaje Java.
- Res: en ella se encuentran todos los recursos utilizados para el desarrollo de la aplicación como son las imágenes, fuente del texto, animaciones, layouts, menús, ...
- Manifests: es el lugar donde se halla el archivo Android Manifest explicado en la sección 2.1.5.

Por otro lado, el proyecto Android también posee una carpeta donde se almacenan los archivos configurables de la compilación Gradle. En esta carpeta se encuentran los archivos *build.gradle* y *settings.gradle* explicados en la sección 2.1.6 del documento.

Android Studio da la posibilidad de visualizar la estructura del proyecto de varias maneras. En la Figura 2.89 se puede observar la estructura de archivos real mediante la forma de vista 'Project' y 'Android' de la aplicación desarrollada:

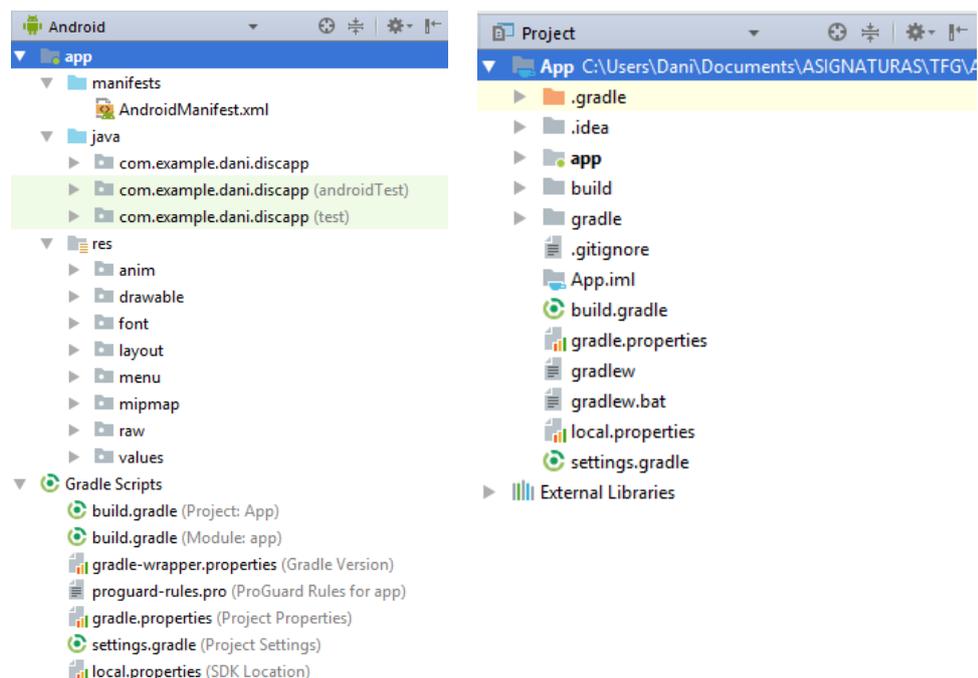


Figura 2.9. Estructura 'Android' (izquierda) y 'Project'(derecha) del proyecto desarrollado

2.2.3 Entorno Java

El primer paso que hay que realizar para comenzar a desarrollar una aplicación para el sistema operativo Android, es instalar en el equipo el kit de desarrollo Java, en inglés *Java Development Kit (JDK)*.

Se debe acceder a la dirección web de Oracle [19] y descargar la última versión del JDK 8.

JDK 8 da la posibilidad, por medio de herramientas útiles, de desarrollar aplicaciones y componentes mediante el uso del lenguaje Java que se ejecutan en la plataforma Java [6]. En la Figura 2.10 se observan las diferentes descargas del JDK 8 posibles en función del sistema operativo.

Kit de desarrollo de Java SE 8u211		
Debe aceptar el Acuerdo de licencia de Oracle Technology Network para Oracle Java SE para descargar este software.		
<input type="radio"/> Aceptar acuerdo de licencia <input checked="" type="radio"/> Negar acuerdo de licencia		
Descripción del producto / archivo	Tamaño del archivo	Descargar
Linux ARM 32 Hard Float ABI	72.86 MB	jdk-8u211-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.76 MB	jdk-8u211-linux-arm64-vfp-hflt.tar.gz
Linux x86	174.11 MB	jdk-8u211-linux-i586.rpm
Linux x86	188.92 MB	jdk-8u211-linux-i586.tar.gz
Linux x64	171.13 MB	jdk-8u211-linux-x64.rpm
Linux x64	185.96 MB	jdk-8u211-linux-x64.tar.gz
Mac OS X x64	252.23 MB	jdk-8u211-macosx-x64.dmg
Solaris SPARC de 64 bits (paquete SVR4)	132.98 MB	jdk-8u211-solaris-sparcv9.tar.Z
Solaris SPARC de 64 bits	94.18 MB	jdk-8u211-solaris-sparcv9.tar.gz
Solaris x64 (paquete SVR4)	133.57 MB	jdk-8u211-solaris-x64.tar.Z
Solaris x64	91.93 MB	jdk-8u211-solaris-x64.tar.gz
Windows x86	202.62 MB	jdk-8u211-windows-i586.exe
Windows x64	215.29 MB	jdk-8u211-windows-x64.exe

Figura 2.10. Descarga de JDK 8

2.2.4 SDK Android

El kit de desarrollo software, en inglés *Software Development Kit* (SDK), es el lugar donde se encuentran todas las herramientas necesarias para llevar a cabo el desarrollo de una aplicación informática en un entorno de trabajo.

Android Studio cuenta con un administrador de SDK llamado *Android SDK Manager*, mediante el cual el programador tendrá acceso a todas las versiones de Android y podrá actualizar las que tiene ya instaladas [6]. Se pueden encontrar las siguientes 3 pestañas de configuración [25]:

- *SDK Platforms*: se puede descargar el nivel de API que se desee de forma completa o simplemente descargando alguna de las herramientas que ofrece. Por otro lado, para poder compilar el código se ha de tener descargado al menos una versión de plataforma Android. La Figura 2.11. muestra la pestaña *SDK Platforms* de *SDK Manager*.

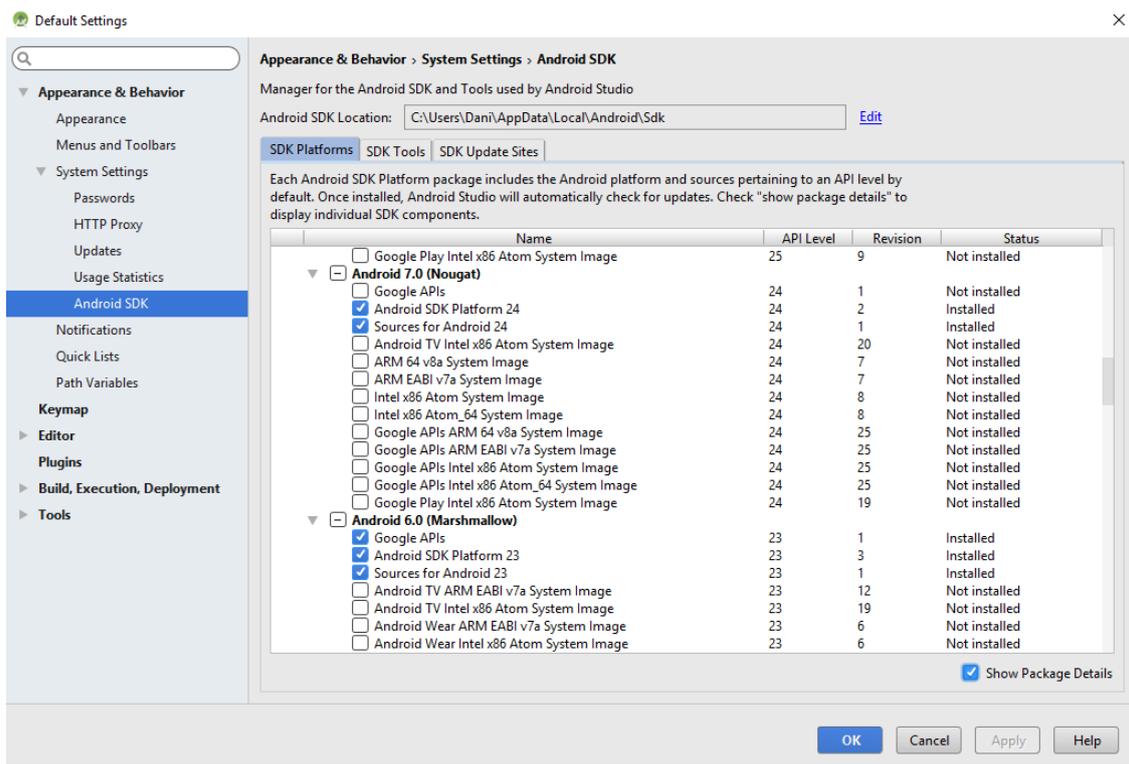


Figura 2.11. Android SDK Manager. SDK Platforms

- **SDK Tools:** incluye cada una de las herramientas necesarias para utilizar las plataformas de Android. La Figura 2.12 muestra la pestaña *SDK Tools* de *SDK Manager*.

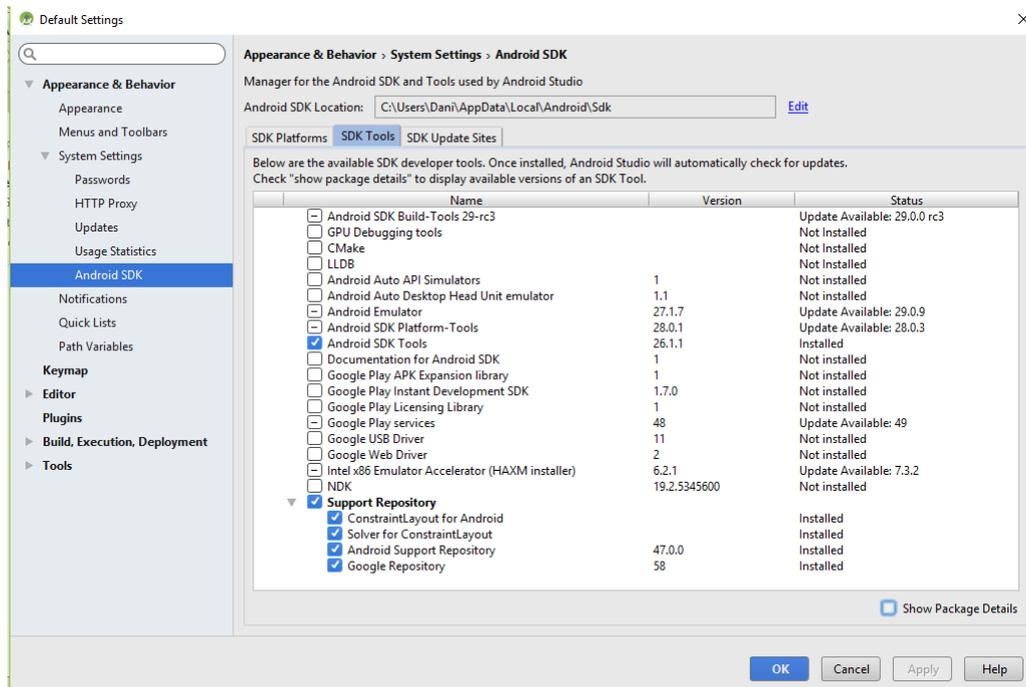


Figura 2.12. Android SDK Manager. SDK Tools

- **SDK Update Sites:** en esta sección aparecerán los lugares donde Android Studio buscará herramientas y actualizaciones de herramientas para en el caso de las haya, notificarlo. También es posible añadir sitios externos a Android Studio que ofrezcan herramientas de desarrollo. Un ejemplo de ello son las bibliotecas de API que pueden ofrecer los fabricantes de dispositivos Android. Simplemente se deberá de agregar la URL de dicho proveedor de bibliotecas. La Figura 2.13 muestra la pestaña *SDK Update Sites* de *SDK Manager*.

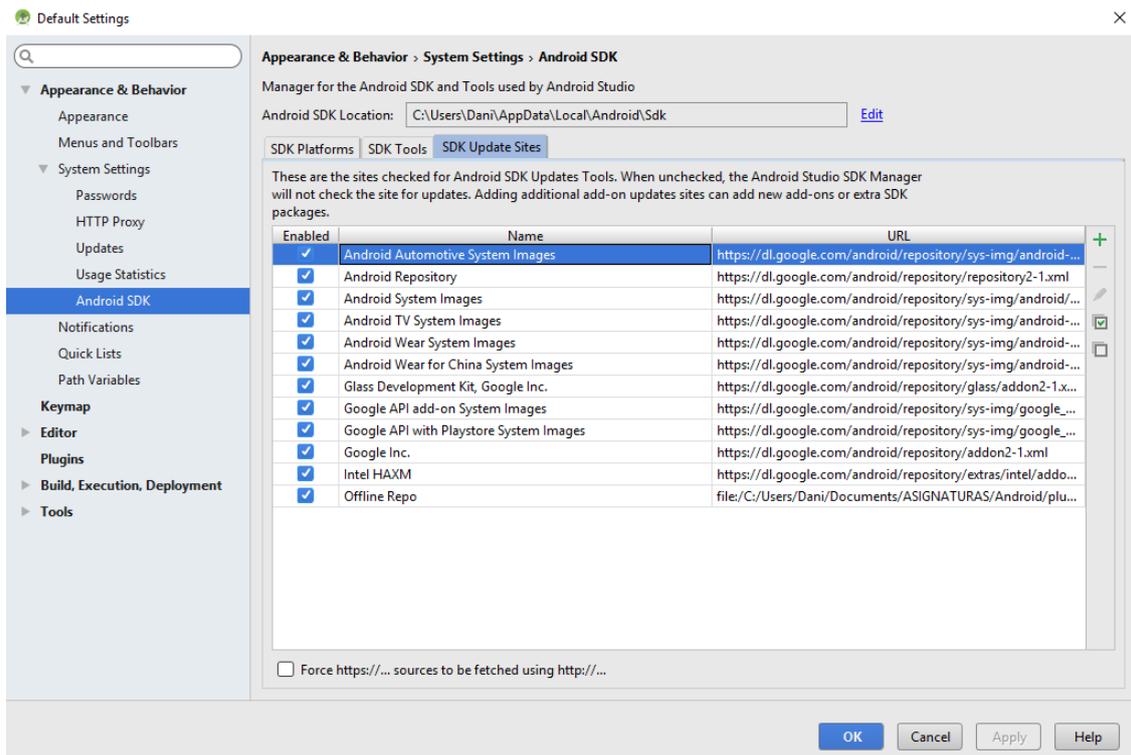


Figura 2.13. Android SDK Manager. SDK Update Sites

Además, se puede visualizar de forma estructurada mediante carpetas en el directorio donde se han almacenado los archivos descargados. Dichos ficheros se distribuirán en las siguientes carpetas [6]:

- *Add-ons*: SDK instalado fuera del SDK principal.
- *Build-tools*: herramientas necesarias para crear y exportar una aplicación.
- *Docs*: documentación de Android.
- *Platform-tools*: se almacenan los archivos propios del desarrollo y depuración de aplicaciones.
- *Platforms*: se guardan las versiones de SDK instaladas ordenadas por número de versión.
- *Samples*: las APIs de Android cuentan con distintos códigos fuente de ejemplo para los programadores. Estos ejemplos se almacenan en la carpeta *Samples*.
- *Sources*: contenedor del código fuente de Android.
- *Tools*: se guardan las distintas herramientas que pueden resultar útiles durante el desarrollo de una aplicación Android.
- *Skins*: contiene las distintas apariencias de los emuladores.

2.2.5 Emulador

El emulador de Android Studio es la herramienta que utilizan los programadores para realizar simulaciones de su implementación de código en dispositivos Android virtuales de distinto tipo, tamaños y resolución. Estos aspectos se pueden configurar desde *Virtual Device Configuration* (VDC) que muestra la Figura 2.14:

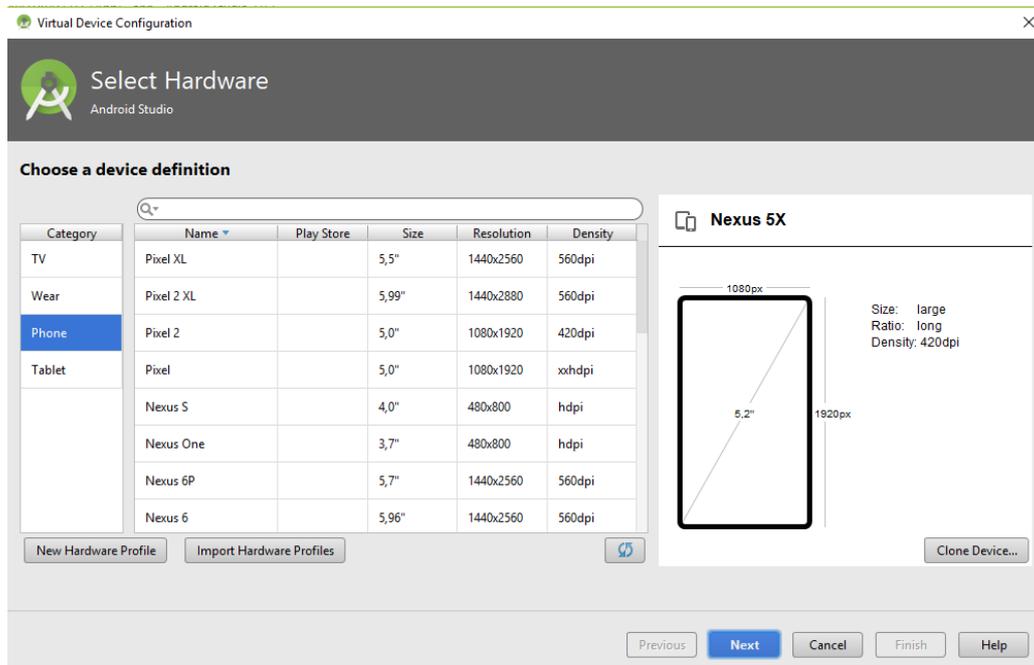


Figura 2.14. Virtual Device ConFiguRation

2.3 SQLite

Es una biblioteca en lenguaje C utilizada como motor de base de datos SQL que se caracteriza por ocupar poco espacio, ser rápida, no depender de factores externos, ser verdaderamente fiable y completo [28].

Según la web oficial de SQLite [28], actualmente es el motor de gestión de bases de datos más utilizado del mundo estando integrado en toda la telefonía móvil, y computadoras y son usadas en aplicaciones de uso diario. Hoy en día son utilizadas como herramientas para almacenar información persistentemente de forma sencilla.

A continuación, se procede a nombrar algunas de las propiedades que caracterizan al motor de bases de datos SQLite y le hacen ser el más utilizado para el desarrollo de aplicaciones Android [29]:

- Proporciona la información necesaria de forma consistente y aislada.
- La configuración y la administración no son necesarias en SQLite.
- Ofrece una gran variedad de funciones para manejar la información almacenada.

- Se trata de un archivo multiplataforma, es decir, puede ser compilado en múltiples plataformas informáticas: iOS, Linux, Solaris, VxWorks, Android, Windows y Mac.
- Admite bases de datos de tamaños como el terabyte.
- Tiene una API de uso fácil.
- Es un motor de bases de datos que tiene una gran rapidez en las transacciones de información.
- Está escrito en ANSI-C, el cual es un estándar para el lenguaje C publicado por el Instituto Nacional Estadounidense (ANSI).
- El código fuente proporcionado está comentado de forma correcta.
- No depende de contenido externo a SQLite.
- Se trata de un servicio de dominio público.
- Cuenta con una interfaz de línea de comandos (CLI), es decir, el usuario puede manejar la base de datos SQLite a partir de una línea de texto simple [30].

En definitiva, el motor de gestión de bases de datos SQLite se define por su capacidad de gestionar bases de datos de gran tamaño, por su estabilidad, por el coste que supone y por el eficiente rendimiento que da a los usuarios.

2.3.1 Diferencias con otros gestores de bases de datos

Existen otros gestores de bases de datos como son SQL Server, Oracle DB o MySQL. Sin embargo, SQLite tiene ciertas ventajas respecto a ellos. Seguidamente se remarcan alguna de las características mencionadas anteriormente que diferencia a SQLite del resto de motores de bases de datos:

- Es de código abierto, el usuario puede acceder a código, instrucciones de escalabilidad y archivos de compilación de forma libre.
- No es necesario realizar una configuración previa.
- Se encarga de implementar una serie de librerías que gestionan la información, es decir, no es requerido el uso de un soporte de un servidor.
- Genera un archivo para el esquema de una base de datos. Con ello, la seguridad de la aplicación aumenta ya que ningún servicio externo puede acceder a la información interna.

Es por ello, y por el resto de las características mencionadas y explicadas previamente, por lo que se ha decidido llevar a cabo una gestión de información en la aplicación desarrollada en el proyecto a través del motor SQLite, un motor de gestión de bases de datos sencillo.

2.3.2 Elementos esenciales

Para el desarrollo de una base de datos en un dispositivo Android se deben contar con tres elementos fundamentales: *SQLiteOpenHelper*, *SQLiteDatabase* y el cursor.

SQLiteOpenHelper es una clase la cual se encarga de gestionar y crear las nuevas versiones de la base de datos. La clase que extienda de *SQLiteOpenHelper* debe contener atributos como son el nombre de la base de datos, el número de versión de la base de datos, el nombre de la/s tabla/s que componen la base de datos, los identificadores de las columnas de la/s tabla/s y la consulta de creación de la tabla. Por otro lado, dispone de tres métodos:

- Un constructor: con el objetivo de crear objetos que faciliten la interacción con la base de datos.
- *onCreate()*: en primer lugar comprueba si la base de datos ya existe mediante el método *checkDatabase()*. Si la base de datos ya ha sido creada, no realiza ninguna acción. En caso de no haber sido creada aún, mediante el método *execSQL()*, se crean las tablas.
- *onUpgrade()*: es el método encargado de gestionar las versiones de la base de datos. Cuando la versión de la base de datos ha cambiado, debe comprobar la compatibilidad de la nueva con la antigua.

Por otro lado, *SQLiteDatabase* es el lugar donde se encuentren todos aquellos métodos a desarrollar para interactuar con la base de datos. Por ejemplo, introducir o borrar información. Debe existir una clase que contenga los métodos necesarios para gestionar la información almacenada en una tabla determinada. En dicha clase, se definen atributos como el nombre de la tabla, el nombre e identificador de cada columna y una variable *SQLiteDatabase* que represente a la base de datos. A continuación, se describen los métodos que se han de implementar para hacer una consulta en la base de datos previamente cargada, actualizar los valores que contiene, insertar una nueva fila o eliminar una fila existente:

- Se debe desarrollar un método público que devuelva un objeto de tal manera que los atributos que lo componen se correspondan con los datos que se obtienen de la consulta. Por ejemplo, si se desea obtener de una determinada tabla los valores correspondientes a las columnas de 'nombre' y 'apellido' de una persona, el objeto que devuelve el método debe estar compuesto por dos atributos de tipo String.

Se le pasa como parámetro una referencia de *SQLiteDatabase* de la base de datos y en el caso de querer obtener una única fila de la tabla en cuestión, un atributo como un código que referencie la fila (por ejemplo, un número).

Mediante el método *query()* se realiza la consulta a la base de datos pasando como parámetros la tabla que se desea consultar, el nombre de las columnas que posee y un identificador de la fila que se quiere.

Dicha consulta es almacenada en una variable cursor. El cursor es el elemento que almacena toda la información de una consulta a la base de datos. Si el cursor está vacío no se realiza ninguna acción, pero si contiene información, se crea un objeto donde se almacene su información.

En el caso de realizar una consulta de toda la tabla y no de una única fila, mediante el método *query()* se obtiene la información de toda la tabla y se almacena en el cursor. Finalmente mediante el método *moveToNext()* se recorre cada fila almacenada en el cursor y se guarda en un ArrayList de objetos.

- Para actualizar el valor de una celda de la tabla de la base de datos se debe crear un método al que se le pase como parámetro un objeto que contenga los valores que se quieren actualizar o directamente una variable que contenga el valor en sí. Se hace uso de un objeto ContentValues, que representa una fila de la tabla, donde mediante el método *put()* se introduce la pareja de valores del nombre de la columna y el valor que se le quiere dar. Finalmente mediante el método *update()*, pasando como parámetro el objeto ContentValues, la tabla y el identificador de la fila, se actualiza la tabla correspondiente.
- Para insertar una nueva fila en la base de datos se hace uso nuevamente de la clase ContentValues. Mediante el método *put()* se agregan los valores de cada columna y finalmente mediante el método *insert()* se inserta la fila nueva en la tabla de la base de datos que se indique.
- Para eliminar una fila de una tabla de la base de datos se ha de llamar al método *delete()* con el nombre de la tabla y la condición que identifica el registro que se elimina.

Finalmente, hay que destacar que hay que seguir cuatro pasos fundamentales para lograr una correcta inserción de datos en la base de datos: en primer lugar, hay que inicializar la base de datos. Seguidamente hay que comenzar una conexión con la base de datos en modo escritura. Posteriormente se insertan los datos con los métodos generados y finalmente se ha de cerrar a conexión.

2.3.3 Inicialización de la base de datos SQLite mediante ficheros .txt

Además del método de inserción de valores en una tabla de la base de datos explicado en la sección 2.3.2., existe la posibilidad de insertar la información a partir de un fichero *.txt* en el que los valores de cada fila estén separados por ‘;’. Dicho fichero *.txt* es más sencillo generarlo a partir del software Excel.

En primer lugar, se ejecuta Excel y se crea la tabla correspondiente a la información que se quiere introducir en la base de datos de la aplicación.



Seguidamente se ha de guardar por ejemplo como archivo *.csv* con el objetivo de separar los valores de cada fila de la tabla sean separados por ‘;’.

En segundo lugar, una vez guardado el archivo como *.csv*, se ha de abrir con un editor de texto y es entonces cuando se puede observar que la información ha sido correctamente separada por ‘;’. A continuación, se debe guardar el archivo abierto en el editor de texto como archivo *.txt*.

El archivo *.txt* se guarda en la carpeta *raw*, dentro de la carpeta de recursos del proyecto *res*. *Raw* es la carpeta en la que se almacenan los recursos que no deben ser procesados.

Finalmente, se procede a crear un método en el lugar donde se desee cargar la información en la base de datos que consistirá en:

1. Mediante un objeto de la clase que desciende de *SQLiteOpenHelper* y el método *getReadableDatabase()* se obtiene el valor de un objeto de la clase *SQLiteDataBase* y con ello una referencia de la base de la base de datos.
2. Pasando como parámetros la referencia de la base de datos y el nombre de la tabla en la que queremos insertar la información al método *queryNumEntries()* de la clase *DatabaseUtils*, se obtienen el número de registros que contiene dicha tabla.
3. En el caso de que la cantidad de registros en la tabla sea cero, querrá decir que aún no ha sido generada la información en ella. De igual forma que en el paso 1 se obtiene una nueva referencia de la base de datos y mediante el método *beginTransaction()* se indica que se va a comenzar a la edición de información que contiene la tabla [32].
4. Se almacena en un objeto *InputStream* el contenido del archivo *.txt*. Una vez generado un objeto *ByteArrayOutputStream* donde guardar la información en un array de bytes, se lee línea a línea el contenido del objeto de la clase *InputStream* y se almacena en este último por medio de *write()*.
5. Se guarda la información en un array de Strings y mediante *split()* y pasando por parámetro ‘;’, es posible detectar la información que estaba fraccionada. Por medio de un objeto *ContentValues* y el método *put()* pasando por parámetros la pareja de clave y valor correspondiente, se introduce la información en la tabla de la base de datos.
6. La llamada al método generado se hará en el momento que el desarrollador precise de la creación de la tabla.



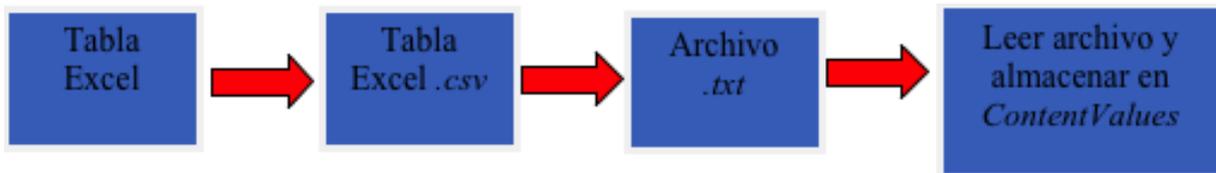


Figura 2.15. Proceso de inserción de datos por medio de tabla Excel

2.4 MeaningCloud

Se trata de una empresa con base en Estados Unidos que ofrece los servicios software necesarios para realizar un análisis semántico de un texto, es decir, extraer de forma automática información de alto valor del texto [67].

Para llevar a cabo este exhaustivo análisis, en primer lugar, se procede a la estructuración del texto a analizar. Posteriormente se descubren distintos patrones en el texto estructurado y por último se interpretan los resultados obtenidos. Se utilizan técnicas de estadística, aprendizaje automático, lingüística computacional y minería de datos. [33]

En el proceso de análisis de textos, el servicio que ofrece MeaningCloud puede suponer algunas de las siguientes tareas [33]:

- Etiquetado gramatical: asigna a cada una de las palabras que componen el texto una categoría gramatical.
- *Clustering*: analiza varios textos detectando las diferencias que existe entre ellos y los clasifica. Se suele realizar esta tarea cuando el objetivo es localizar textos duplicados en un documento.
- Clasificación: se le asigna una o varias categorías al texto analizado. De esta manera, se conoce el tema global del texto.
- Extracción de información: detecta nombres de personas, empresas, lugares, conceptos abstractos, ... Así, el analizador de textos es consciente de aquellas palabras que proporcionan una mayor información.
- Análisis de sentimiento: detecta la polaridad de un texto (muy positiva, positiva, neutra, negativa, muy negativa o sin polaridad).

2.4.1 Ventajas de los analizadores de texto automáticos

A continuación, se definen las principales ventajas del uso de los analizadores semánticos automáticos [33]:

- El análisis semántico que es realizado por varias personas de forma independiente no siempre puede llegar a ser el mismo ya que cada uno tiene sus propios criterios. Los analizadores de textos automáticos, como el de

MeaningCloud, a pesar de que el grado de precisión no sea exactamente el 100%, el criterio y los procedimientos que sigue son siempre los mismos.

- El tiempo que tarda en realizar un análisis de un texto es del orden de milisegundos.
- La disponibilidad de la herramienta es absoluta en cualquier momento, es decir, la dependencia de otra persona para realizar el análisis no existe.
- Si el analizador de textos automático se adapta de forma correcta al contexto, se obtienen unos grados de precisión y cobertura paralelos a los del procesamiento humano.

2.4.2 APIs disponibles en MeaningCloud

En esta sección del documento se describen las APIs que ofrece MeaningCloud y de las cuales se pueden realizar pruebas desde su página web [34]:

- Categorización profunda: mediante reglas morfológicas y semánticas, se realiza una clasificación por categorías de un texto determinado. Además, un texto puede ser clasificado en una o varias categorías.
- Temas de extracción: extrae la información de mayor valor como conceptos abstractos, entidades, fechas, cantidades, ...
- Análisis de los sentimientos: detecta polaridad, ironía, subjetividad y acuerdo emocional de un texto.
- Identificación de idioma: detecta el idioma en el que está escrito el texto.
- Agrupación de texto: agrupa los documentos que son similares y detecta los temas más significativos.
- Resumen: realiza un resumen de un texto determinado.
- Análisis de la estructura del documento: detecta las diferentes secciones de un documento: título, secciones y resumen.

2.4.3 Ejemplos de uso de análisis de textos automático

A continuación, se detallan algunos ejemplos donde se hace uso de los análisis semánticos de textos [33]:

- Las empresas utilizan este sistema para conocer más a fondo la opinión de sus clientes (voz del cliente). Por ejemplo, se analizan las respuestas en encuestas.
- Se llevan a cabo análisis semánticos de los medios sociales tanto tradicionales como nuevos. Se analiza la información que crean los informadores y la información que generan los clientes.



- Al igual que se realiza un análisis semántico de la voz del cliente, también se realiza del empleado (voz del empleado).
- En los casos de investigación científica en los que es necesario llevar a cabo la lectura comprensiva de libros de gran volumen y gran cantidad de documentos, se realiza un análisis semántico realizar una recolección de información más sencilla.
- Los medios de comunicación realizan análisis de su audiencia para posteriormente generar un contenido personalizado de mayor calidad y de forma más rápida.
- Es utilizado por organizaciones de salud o judiciales para llevar a cabo una clasificación de expedientes.

2.5 Firebase

Envolve fue una empresa emergente de desarrollo web que fue creada por Andrew Lee y James Tamplin en 2012 con el principal objetivo de proporcionar un sistema de chat en sus páginas web. Tras comprobar que los programadores utilizaban el sistema chat para pasar paquetes de información de sus aplicaciones, los fundadores tomaron la decisión de generar un sistema de chat y un sistema de arquitectura en tiempo real de forma independiente. De esta manera, se propició la fundación de Firebase [35].

En octubre de 2014 Google adquiere Firebase y desde entonces no ha parado de crecer como plataforma de desarrollo móvil y desarrollo web ofreciendo servicios de gran potencial gracias a la adquisición de plataformas informáticas como Divshot o Crashlytics [35].

Aplicaciones muy conocidas como el analizador sonoro Shazam o el periódico The New York Times hacen uso de algunos de los servicios ofrecidos por Firebase. Siguiendo la estela de grandes aplicaciones y conociendo las herramientas que propone Firebase, las cuales se detallan a continuación, se ha tomado la decisión de impulsar el potencial de la aplicación del proyecto con algunos de los servicios que se ofrecen.

2.5.1 Servicios que ofrece Firebase

Entre todos los servicios que ofrece la plataforma informática de Firebase, se pueden clasificar en tres grupos con un gran objetivo cada uno de ellos: la compilación de aplicaciones potentes, mejorar la calidad de las aplicaciones y generar crecimiento en el negocio [36].

- Compilar aplicaciones más potentes. Los servicios que ofrece Firebase permiten compilar mejores y más seguras aplicaciones mediante las siguientes herramientas [36]:



- Cloud Firestore: por medio de una base de datos NoSQL (no usa SQL como lenguaje principal [37]) en la nube, esta herramienta sincroniza y consulta datos de manera eficaz, además de contener un servicio para cuando el usuario no tenga conexión a internet.
- Cloud Functions: pone a disposición de los desarrolladores funciones para personalizar el código.
- Authentication: ofrece un módulo de autenticación para que la experiencia de los usuarios con la aplicación sea más segura.
- Hosting: herramientas para llevar a cabo un desarrollo web moderno y sencillo.
- Cloud Storage: posibilita la carga y descarga de archivos generados por el usuario, sin tener en cuenta el estado o calidad de la red.
- Realtime Database: proporciona lo necesario para llevar a cabo guardados y consultas de información a una base de datos NoSQL alojada en la nube en tiempo real.
- Mejorar la calidad de las aplicaciones: entre sus servicios, Firebase ofrece también las herramientas necesarias para que el desarrollador tenga en cuenta datos estadísticos y de rendimiento para que utilice sus recursos de la manera más eficaz posible [36].
 - Crashlytics: ofrece una manera sencilla de solucionar los problemas de ejecución de la aplicación. En el caso de que haya muchos fallos, Firebase proporciona al desarrollador una lista de los problemas la cual es fácil de manejar. Además, genera alertas en tiempo real para que el programador esté prevenido. Finalmente, la herramienta Crashlytics proporciona datos estadísticos sobre qué problema de la posible lista de errores debe solucionar primero el desarrollador.
 - Test lab: el dueño de la aplicación es capaz de ejecutar de forma personalizada su código mediante un simulador virtual de Google.
 - Supervisión del rendimiento: genera un informe sobre el rendimiento que ha tenido la aplicación en los dispositivos de los usuarios.
- Generar crecimiento en el negocio: mediante distintas herramientas o servicios, Firebase fomenta que los usuarios que descargan la aplicación desarrollada la mantengan instalada en sus dispositivos y la utilicen [36].
 - In-App Messaging: Firebase genera mensajes personalizables para animar al usuario a que siga utilizando la aplicación desarrollada.
 - Google Analytics: permite la visualización en gráficas de la actividad de los usuarios, en qué pantallas interactúan más, los ingresos que genera la aplicación, la forma en la que se están adaptando los usuarios a la aplicación y de qué región son los usuarios que se conectan a la aplicación.



- Predictions: en función de la forma de utilizar la aplicación, Firebase genera unos datos estadísticos y a partir de ellos la herramienta Predictions puede crear grupos o segmentos de usuarios. Con ello, el desarrollador tiene una idea del tipo de audiencia que tiene y puede modificar la aplicación a su gusto.
- Cloud Messaging: es un sistema que permite el envío de mensajes entre usuarios de la aplicación.
- Remote Config: permite que cada usuario tenga una presentación distinta en su dispositivo.
- Dynamic Links: permite que el usuario pueda abrir cualquier sitio web dentro de la aplicación y que desde el sitio web de la aplicación pueda acceder a una sección de la propia aplicación entre otros servicios.

La plataforma de desarrollo web y móvil Firebase puede ser utilizado en iOS, Android, web, C++ y Unity. A continuación, en la Tabla 2.2 se puede observar qué servicios que están disponibles por cada uno de los lenguajes de programación:

iOS	Android	Web	C++	Unity
Cloud Firestore	Cloud Firestore	Cloud Firestore	-	-
Cloud Functions	Cloud Functions	Cloud Functions	Cloud Functions	Cloud Functions
Authentication	Authentication	Authentication	Authentication	Authentication
-	-	Hosting	-	-
Cloud Storage	Cloud Storage	Cloud Storage	Cloud Storage	Cloud Storage
Realtime Database	Realtime Database	Realtime Database	Realtime Database	Realtime Database
Crashlytics	Crashlytics	-	-	Crashlytics
Supervisión rendimiento	Supervisión rendimiento	-	-	-
Test Lab	Test Lab	-	-	-
Google Analytics	Google Analytics	-	Google Analytics	Google Analytics



Predictions	Predictions	-	Predictions	Predictions
In-App Messaging	In-App Messaging	-	-	-
Cloud Messaging	Cloud Messaging	Cloud Messaging	Cloud Messaging	Cloud Messaging
Remote Config	Remote Config	-	Remote Config	Remote Config
Dynamic Links	Dynamic Links	Dynamic Links	Dynamic Links	Dynamic Links

Tabla 2.2. Servicios ofrecidos por Firebase dependiendo del lenguaje

2.5.2 Agregar Firebase a proyecto Android:

Android Studio cuenta con el asistente de Firebase (*Firestore Assistant*) mediante el cual el programador puede agregar los servicios que se necesiten de Firebase al proyecto Android que se está desarrollando [38].

En primer lugar, una vez abierto el proyecto, desde la pestaña de herramientas (*Tools*) de Android Studio se ha de abrir el asistente de Firebase (*Tools>Firestore*) tal y como se muestra en la Figura 2.16.

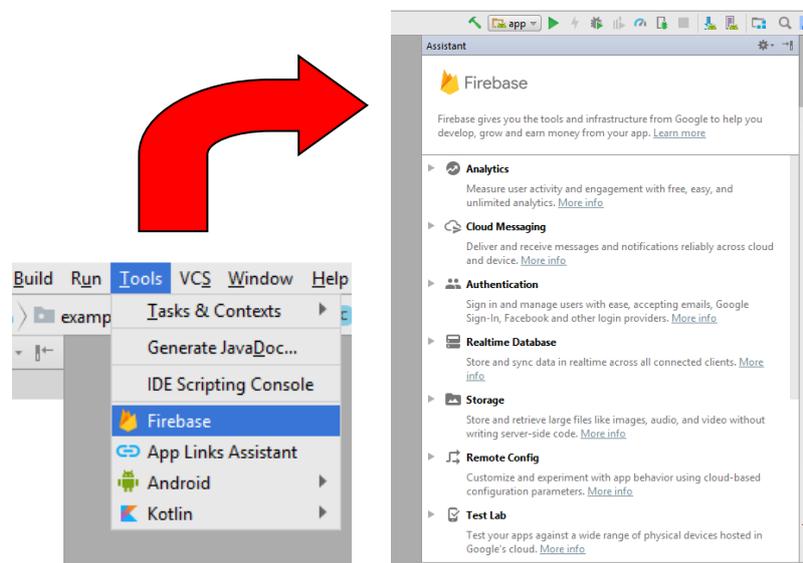


Figura 2.16. Abrir asistente de Firestore.

A continuación, cuando el servicio que queremos utilizar de Firestore está elegido, se debe abrir su pestaña correspondiente y pinchar en el enlace que

contiene. Seguidamente, haciendo clic en ‘Conectar a Firebase’, se añade de forma automática el código necesario para que la aplicación que se está desarrollando esté enlazada con el proyecto Firebase.

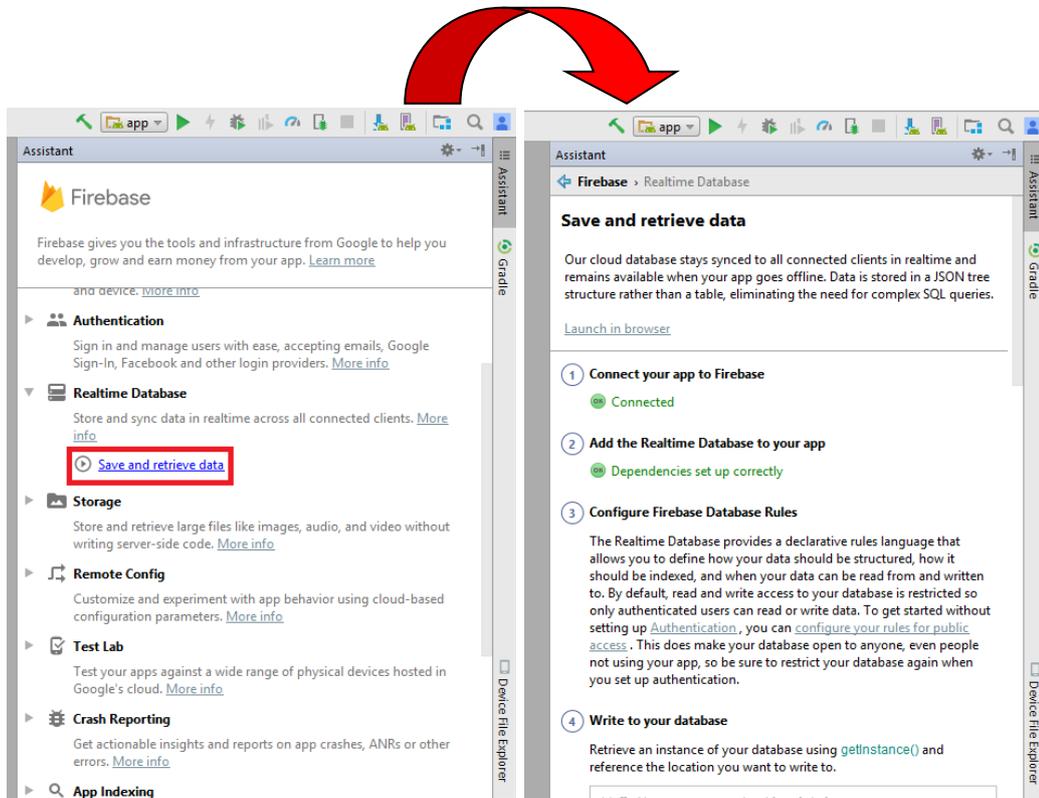


Figura 2.17. Asistente de Firebase

Una vez que la aplicación del desarrollador esté conectada con el servicio correspondiente de Firebase, se ha de comprobar que la versión del complemento *Google Services* del archivo *build.gradle* a nivel de proyecto es la más reciente y que las versiones de las bibliotecas de Firebase para Android del archivo *build.gradle* a nivel de módulo sean también las más recientes. Posteriormente, sincroniza la aplicación para que las dependencias tengan las versiones necesarias.

Finalmente se ha de realizar la configuración del proyecto Firebase desde la consola la página web [39] y ejecutar la aplicación para que Firebase sea consciente de que el servicio ha sido integrado de forma correcta [38].

Por otro lado, existe una segunda forma de conectar la aplicación que se está desarrollando en Android a la plataforma Firebase. Se trata de agregar los SDK de los servicios que se vayan a utilizar al proyecto Android de forma manual [38].

En primer lugar, en el archivo *build.gradle* de nivel de aplicación del proyecto, se ha de agregar las dependencias a los SDK que se vayan a utilizar. Todas las bibliotecas disponibles para los servicios que ofrece Firebase se pueden encontrar en la sección ‘Bibliotecas disponibles’ de [40]. Por ejemplo:

```
implementation 'com.google.firebase:firebase-database:11.0.4'
```

Seguidamente se ha de sincronizar el proyecto. De esta manera se comprueba que las dependencias de los SDKs añadidas en el paso anterior tienen la versión más reciente.

Finalmente, tras haber configurado el proyecto Firebase desde la consola disponible en la web [39], se ejecuta el proyecto para enviar a Firebase la verificación de que la conexión con su plataforma ha sido correcta.

2.5.3 Sintetizador TTS (*Text To Speech*)

Se trata de un módulo en el que se lleva a cabo un sintetizado de texto para que o bien sea reproducido o bien sea grabado el audio. Es decir, cualquier texto escrito en formato digital es reproducido por el dispositivo.

En la actualidad los sintetizadores de texto a voz son utilizados por aplicaciones que son utilizadas por un gran número de usuarios. Por ejemplo, las aplicaciones de navegación GPS en tiempo real suelen disponer de este servicio con el objetivo de que si el usuario no puede atender las indicaciones que se muestran en la pantalla del dispositivo, pueda continuar recibiendo instrucciones para llegar a su destino. Otro ejemplo que utiliza dicho servicio es el traductor de Google. Tras realizar una consulta de traducción de un texto, la aplicación Traductor de Google pone a disposición del usuario un botón mediante el cual se reproducirá el texto traducido.

En definitiva, añadir un sintetizador de texto a voz en una aplicación puede ser de gran utilidad a los usuarios. Hay personas con discapacidad intelectual que pueden llegar a tener algunos problemas de vista si el texto es muy pequeño. El objetivo de agregar esta funcionalidad a la aplicación es que los usuarios que la utilicen se sientan cómodos con ella y no tengan que hacer grandes esfuerzos para usarla.

2.5.4 Módulos de síntesis de texto a voz

El proceso del sintetizador de texto a voz está compuesto por varios módulos que se organizan según indica la siguiente Figura 2.18 [41]:

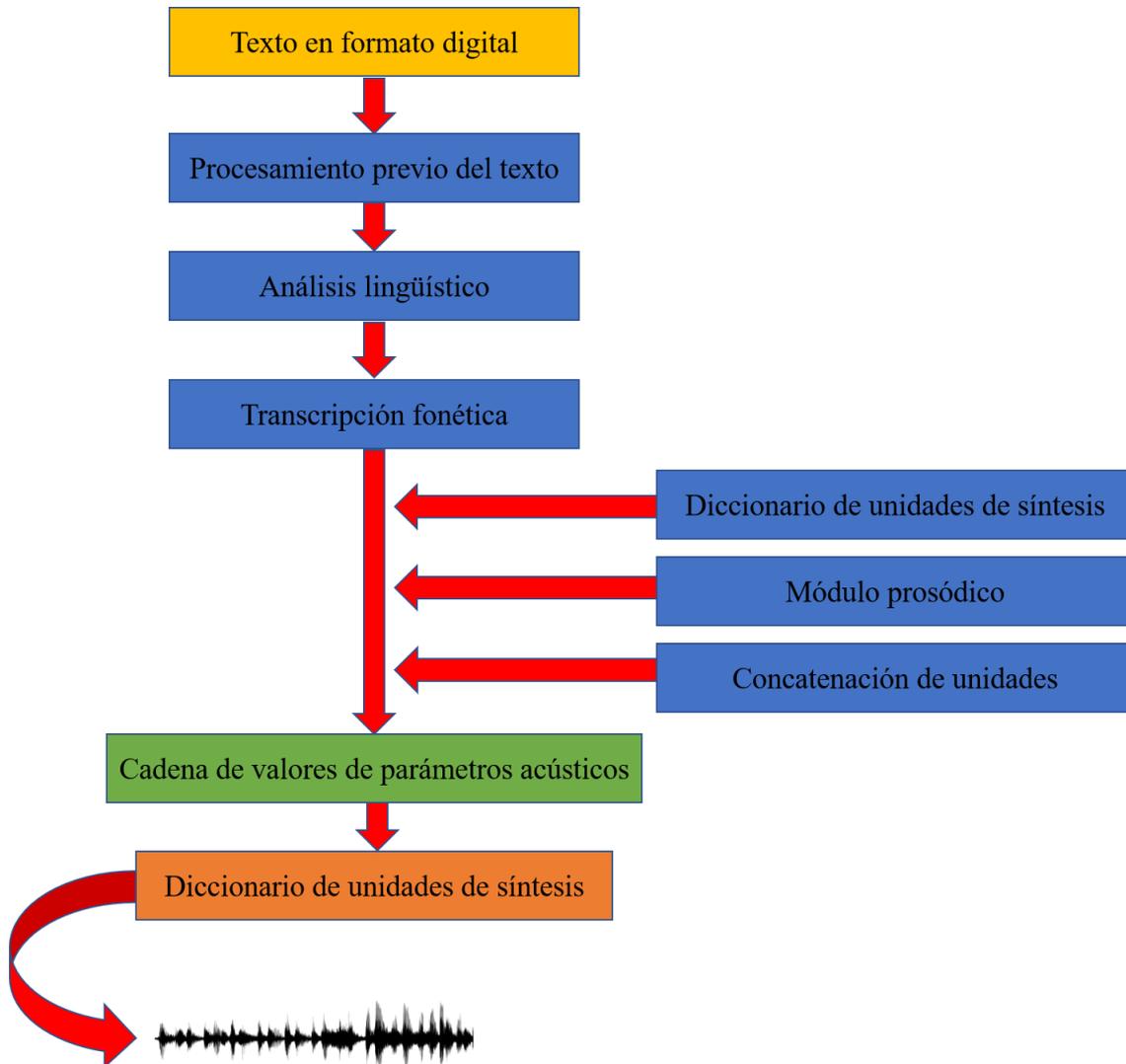


Figura 2.18. Módulos de síntesis de texto a voz

- Procesamiento previo del texto: sistema para la lectura del texto en el caso de que contengan signos de puntuación sin valor lingüístico, es decir, horas, fechas, siglas, etc.
- Análisis lingüístico: se definen las reglas para el procesamiento sintáctico y morfológico. No siempre aparece este módulo en los conversores de texto a voz.

- Transcripción fonética automática: se trata de un algoritmo que convierte el texto de formato digital en una representación expresada mediante símbolos fonéticos.
- Diccionario de unidades de síntesis: se realiza una selección de unidades de longitud variable recogidas en un inventario de fonemas establecido (diccionario de unidades de síntesis).
- Módulo prosódico: establece la duración de cada una de las unidades del texto, su intensidad, la intensidad global del texto, el nivel del texto y los lugares donde incluir pausas y su duración.
- Concatenación de unidades: se procede a encadenar las segmentaciones del texto.
- Cadena de valores de parámetros acústicos: se define la configuración acústica de cada segmento.
- Módulo de síntesis: finalmente se procede a la transformación de la información de la secuencia de todas las unidades y de la configuración prosódica en una onda sonora.

2.5.5 TTS en Android

Android cuenta con la clase `TextToSpeech` la cual ofrece una gran cantidad de parámetros configurables para que el desarrollador de la aplicación sea capaz de añadir un sintetizador de texto a voz lo suficientemente eficaz. Importando el paquete `'android.speech.tts.TextToSpeech'` se tiene acceso a las clases anidadas, constantes y los principales métodos públicos que se detallan en las Tablas 2.3, 2.4 y 2.5 respectivamente:

Clases anidadas	Descripción
<code>TextToSpeech.Engine</code>	Contiene las constantes y nombres para controlar el paso de texto a voz
<code>TextToSpeech.EngineInfo</code>	Proporciona información sobre el motor de texto a voz instalado en el dispositivo
<code>TextToSpeech.OnInitListener</code>	Una vez inicializada la síntesis de texto a voz, se invocará este método.

Tabla 2.3. Clase `TextToSpeech`. Clases anidadas



Constantes	Tipo	Descripción
ACTION_TTS_QUEUE_PROCESSING_COMPLETED	String	Se ha completado el proceso de síntesis de texto a voz
ERROR	int	Ha ocurrido un fallo
ERROR_INVALID_REQUEST	int	Fallo producido por una solicitud errónea
ERROR_NETWORK	int	Error producido por un fallo en la conectividad
ERROR_NETWORK_TIMEOUT	int	Error producido por un exceso de espera de la red
ERROR_NOT_INSTALLED_YET	int	Fallo producido por la descarga no terminada de los datos de voz
ERROR_OUTPUT	int	Error producido en la salida del sintetizador
ERROR_SERVICE	int	Error en el servicio de TTS
ERROR_SYNTHESIS	int	Fallo producido en el proceso de sintetizar el texto
LANG_AVAILABLE	int	El lenguaje está disponible
LANG_COUNTRY_AVAILABLE	int	El lenguaje está disponible para la configuración regional
LANG_COUNTRY_VAR_AVAILABLE	int	El lenguaje está disponible para el país escogido y para las variantes
LANG_MISSING_DATA	int	Ausencia de datos de idioma
LANG_NOT_SUPPORTED	int	Idioma escogido no compatible



QUEUE_ADD	int	Se añade a la cola una nueva entrada
QUEUE_FLUSH	int	Se eliminan todas las entradas de la cola y se introduce una nueva entrada
STOPPED	int	Pausa solicitada por el cliente
SUCCESS	int	La conversión de texto a voz se desarrolló con éxito

Tabla 2.4. Clase TextToSpeech. Constantes

Métodos	Descripción
<i>addEarcon()</i>	Establece un archivo de sonido y la cadena de texto
<i>addSpeech()</i>	Asigna una cadena de texto y un recurso de sonido a un paquete
<i>areDefaultsEnforced()</i>	Comprueba si la configuración que el usuario tiene en su teléfono móvil es compatible con la aplicación
<i>getAvailableLanguages()</i>	Devuelve los idiomas disponibles para el TTS
<i>getDefaultEngine()</i>	Escoge un motor de síntesis por defecto
<i>getDefaultVoice()</i>	Devuelve la voz predeterminada para el idioma predeterminado
<i>getEngines()</i>	Devuelve todos los motores instalados en el dispositivo
<i>getMaxSpeechInputLength()</i>	Devuelve el tamaño máximo que puede tener una cadena de texto en la entrada
<i>getVoice()</i>	Devuelve una instancia de la voz escogida
<i>isLanguageAvailable()</i>	Especifica si un idioma o lengua está disponible
<i>isSpeaking()</i>	Comprueba si el TTS está hablando
<i>setAudioAttributes()</i>	Define los parámetros de audio que se utilizan en la conversión de texto a voz
<i>setLanguage()</i>	Define la lengua o idioma
<i>setOnUtteranceProgressListener()</i>	Se establecen los escuchadores que recibirán varios eventos relacionados con la síntesis del habla



<i>setPitch()</i>	Define el tono de voz utilizado
<i>setSpeechRate()</i>	Establece la velocidad con la que se habla
<i>setVoice()</i>	Define la voz que se utiliza
<i>shutdown()</i>	Se liberan los recursos utilizados en el proceso del TTS
<i>speak()</i>	Se reproduce el sonido determinado
<i>stop()</i>	Se pausa la reproducción del sonido
<i>synthesizeToFile()</i>	Sintetiza el texto que contiene un archivo específico

Tabla 2.5. Clase TextToSpeech. Métodos públicos sin sus parámetros de entrada

Para concluir esta sección del documento, se recuerda que es importante hacer una llamada al método *shutdown()* al finalizar el uso de la instancia TextToSpeech con el objetivo de liberar los recursos utilizados por el motor de síntesis de texto a voz TextToSpeech.

2.6 Discapacidad intelectual

La Organización Mundial de la Salud (OMS) es un organismo fundado en 1984 el cual tiene como principal objetivo lograr el máximo grado de salud en las personas, tal y como dice la Constitución (*‘un estado de completo bienestar físico, mental y social’*) [57]. Dicho organismo define persona con discapacidad intelectual como *“la persona pueda tener dificultades para comprender, aprender y recordar cosas nuevas, que se manifiestan durante el desarrollo, y que contribuyen al nivel de inteligencia general, por ejemplo, habilidades cognitivas, motoras, sociales y de lenguaje”* [52].

Como se ha comentado en la sección 1.1, la discapacidad intelectual se caracteriza principalmente porque se suele dar entre el nacimiento y los 18 años, el individuo tiene un coeficiente intelectual inferior a una puntuación de 70 y tiene dificultades para realizar actividades con cierto grado de autonomía.

Se observa en la Figura 1.2 de la sección 1.1 del documento, que la discapacidad intelectual es la discapacidad que más probabilidad tiene de concurrir en el grupo de edad de los 0 a los 34 años. Se ha representado este rango de edad ya que la discapacidad intelectual se suele desarrollar desde el nacimiento hasta los 18 años. Con estos datos, organizaciones de ayuda a la discapacidad intelectual han estimado que aproximadamente 1% de la población española posee discapacidad intelectual, lo que supondría 467.200 personas [3].

Actualmente existen planes para personas con discapacidad intelectual o cognitiva que consisten en tratar aspectos como su integración en la sociedad a la vez que ésta se adapta a las necesidades de los individuos, la normalización, el derecho a que

personas con discapacidad intelectual sean atendidas cada una de manera distinta en función de sus síntomas, el derecho a la accesibilidad a determinados servicios y el derecho a mejorar su desarrollo [57].

2.6.1 Diagnóstico

El coeficiente o cociente intelectuales (CI) es una de las formas que existe para evaluar la capacidad mental para aprender, razonar y resolver problemas. Según DSM-IV [58], en la edición revisada, la Asociación Americana de Psiquiatría (APA, 2000), se propone una clasificación de la discapacidad intelectual en función de la puntuación del coeficiente intelectual. Si se encuentra entre 70 y 75 o menor en una prueba de inteligencia indica que el funcionamiento intelectual de la persona es limitado.

Por otro lado, *Diagnosics and Statistical Manual of Mental disorders (DSM-V)* [1] la Asociación Americana de Psiquiatría (APA, 2014) define la discapacidad intelectual como el trastorno que se da durante la fase de desarrollo y que propicia limitaciones en el funcionamiento intelectual (razonamiento, resolución de problemas, aprendizaje académico, ...) y comportamiento adaptativo en dominios conceptuales, sociales y prácticos (comunicación, participación social, la vida en el hogar y en el trabajo, ...).

DSM-V clasifica los trastornos mentales propiciando las descripciones pertinentes para que posteriormente puedan ser diagnosticados y estudiados. El tipo de discapacidad determinará la capacidad que tiene la persona para vivir en la sociedad, dominar ciertas habilidades, trasladarse o no a lugares desconocidos y aprender un oficio. Según el quinto volumen de DSM [1], las discapacidades intelectuales se clasifican de la siguiente manera:

- **Discapacidad Intelectual leve:**
 - **Dominio conceptual:** durante la etapa preescolar de los niños, suele pasar desapercibido y en edades escolares o en adultos se pueden observar dificultades en tareas de lectura, escritura, aritmética, el tiempo y el dinero. En adultos se acentúa las dificultades en la planificación, definición de estrategias a seguir para llevar a cabo ciertas actividades y la memoria.
 - **Domino social:** con respecto a los grupos de edad de desarrollo similar, el individuo tiene problemas a la hora de relacionarse socialmente. Se encuentran dificultades a la hora de comunicarse, mantener conversaciones y en ocasiones, no tienen control de las emociones.
 - **Dominio práctico:** en cuanto al cuidado personal, el individuo se comporta de igual manera que los grupos de edad de desarrollo similar. Sin embargo, necesitan ayuda en ciertas tareas de la vida cotidiana como es la compra, el transporte, la preparación de alimentos, toma de decisiones sobre el cuidado de su salud, etc.



- Discapacidad Intelectual moderada:
 - Dominio conceptual: en la fase de preescolar, el lenguaje y las habilidades pre-académicas se desarrollan de forma lenta. En edades escolares, el desarrollo es considerablemente más lento que los grupos de desarrollo de misma edad en tareas como la lectura, la escritura, las matemáticas, el tiempo y el dinero. Los adultos necesitan ayuda continua diaria para llevar a cabo las actividades de la vida cotidiana.
 - Dominio social: el individuo presenta problemas notorios en el ámbito de la comunicación y comportamiento social. En ocasiones no son capaces de percibir ciertas señales sociales.
 - Dominio práctico: el individuo por medio de un largo proceso de aprendizaje puede llevar a cabo tareas como comer, vestirse e higiene. En ocasiones debe haber alguien que le recuerde lo que tiene que hacer. De igual forma, tras un largo proceso de aprendizaje y apoyo por parte de alguien, pueden llegar a desarrollar las tareas domésticas.

- Discapacidad Intelectual grave:
 - Dominio conceptual: el individuo tiene poca capacidad de comprensión del lenguaje escrito, conceptos numéricos, del tiempo y del dinero. Precisan de ayuda constante para resolver los problemas de la vida cotidiana que surgen.
 - Dominio social: tienen problemas notables en el vocabulario y la gramática. Presentan grandes dificultades para comunicarse. El lenguaje es más utilizado para comunicarse socialmente que para realizar explicaciones.
 - Dominio práctico: la persona necesita ayuda para realizar todas las actividades de la vida cotidiana, es decir, una supervisión constante. En la vida adulta, también se necesita apoyo y ayuda constante para realizar las tareas domésticas.

- Discapacidad intelectual profunda:
 - Dominio conceptual: pueden adquirir habilidades visuoespaciales como la clasificación basada en características físicas, pero presentan graves alteraciones motoras y sensitivas.
 - Dominio social: se comunica mediante símbolos y gestos de manera muy limitada. El individuo es capaz de comprender instrucciones o gestos sencillos y expresa sus deseos mediante comunicación no verbal y no simbólica.
 - Dominio práctico: necesitan ayuda y apoyo constante en todas las actividades y tareas que realizan.



2.6.2 Causas

En un individuo con discapacidad intelectual leve se puede observar diversos factores que interactúan entre sí, como son los genéticos, sociales, ambientales y educacionales. Por otro lado, en muchos casos de discapacidad intelectual moderada y grave se detectan condiciones patológicas que impiden el correcto funcionamiento intelectual [59].

A continuación, se hace una clasificación de las principales causas o factores que influyen en la discapacidad intelectual: factores orgánicos y factores ambientales [59].

Factores orgánicos	Factores prenatales	<ul style="list-style-type: none"> • Infecciones • Drogas y medicamentos • Alteración en la placenta • Edad de la madre • Trastornos funcionales • Diabetes, hipertiroidismo, ... • Alteración en el desarrollo del embrión
	Factores perinatales	<ul style="list-style-type: none"> • Trastornos neonatales: meningitis (inflamación de las meninges), hemorragias intracraneales o anoxia (falta de oxígeno en las células, los órganos o la sangre [47]) • Trastornos intrauterinos: nacimiento prematuro o posmaduro, placenta previa (obstrucción del cuello uterino de la madre [49]) y hemorragias.
	Factores postnatales	<ul style="list-style-type: none"> • Infecciones • Trastornos degenerativos • Malnutrición • Traumatismos craneales • Trastornos desmielizantes (afecta al recubrimiento que se encarga de proteger los nervios del cerebro, la médula espinal y los nervios ópticos [48])
Factores ambientales	<ul style="list-style-type: none"> • Desventaja social 	

	<ul style="list-style-type: none"> • Privación de estímulos en los sentidos. Por ejemplo, privar de visión o de la audición [50] • Abuso infantil • Negligencia infantil
--	---

Tabla 2.6. Factores que influyen en la discapacidad intelectual

2.6.3 Cuadro clínico asociado a la discapacidad intelectual

A continuación, se muestra en la Tabla 2.7. se mencionan las principales ideas de algunos de los trastornos que dan pie a la discapacidad intelectual [52]:

Síndrome de Down	<ul style="list-style-type: none"> • Se trata de la copia adicional del cromosoma 21 del ser humano. • Diagnóstico: análisis cromosómico prenatal y postnatal. • Da lugar a la discapacidad intelectual leve • Retraso ponderal: el niño no alcanza el desarrollo esperado [51]. • Sufren diversos problemas médicos.
Síndrome frágil X	<ul style="list-style-type: none"> • Forma más común de la discapacidad intelectual hereditaria • Ausencia de la proteína necesaria para el desarrollo del cerebro [53]. • Las personas con síndrome X frágil tienen dificultades cognitivas, problemas a la hora de realizar ciertas actividades o tareas, problemas de memoria y dificultad en habilidades matemáticas. • El desarrollo del lenguaje cuando se es niño es más lento. • El individuo trastornos de comportamiento y emocionales. • Además, puede acabar desencadenando en el síndrome del autismo.
Fenilcetonuria	<ul style="list-style-type: none"> • Es el trastorno por medio del cual el cuerpo no es capaz de metabolizar la fenilalanina debido a la falta de la enzima fenilalanina hidroxilasa [54]. • El exceso de fenilalanina en el cerebro provoca daños cerebrales graves • En el nacimiento del individuo no se presentan síntomas. Sin embargo, posteriormente se presentan dificultades en el desarrollo de la persona y piel, cabello y ojos de color más claro.



	<ul style="list-style-type: none">• Presentan problemas motores, de conducta y en ocasiones desemboca en el trastorno del autismo.• Mediante una intervención temprana se puede evitar un daño cerebral de nivel grave.
Hipotiroidismo congénito	<ul style="list-style-type: none">• Ausencia de la hormona tiroidea.• Es de carácter transitoria o permanente.• Carácter transitorio: debido a la falta de yodo.• Carácter permanente: por la dificultad de metabolizar la hormona tiroidea o por ausencia de la hormona estimulante de la tiroides.• El individuo disminuye la ingesta de alimentos, duerme en exceso, tiene un llanto atípico, se constipan con facilidad, tiene baja temperatura corporal y reflejos lentos.• Si no se trata, provoca discapacidad intelectual permanente y problemas respiratorios.• El tratamiento del hipotiroidismo congénito puede llevar varios años. El tratamiento del permanente es de por vida.
Síndrome de Prader-Willi	<ul style="list-style-type: none">• Es poco frecuente.• Afecta al cromosoma 15q11-13• Sus síntomas son: tono muscular débil, estatura baja, desarrollo sexual no completo, dificultades en la ingesta, dificultades cognitivas y una sensación de hambre constante que puede dar lugar a la obesidad.• Se detecta mediante pruebas genéticas.• No tiene cura.• Si se detecta temprano, se lleva a cabo una intervención óptima.
Síndrome de Angelman	<ul style="list-style-type: none">• La madre deja de contribuir con una región del cromosoma 15.• El individuo presenta dificultades en el lenguaje, retraso en el desarrollo, discapacidad intelectual, comportamiento alegre y falta de control muscular [55].• Se diagnostica mediante las características clínicas mencionadas en el punto anterior, resultados de pruebas genéticas y análisis citogenético (análisis de cromosomas).• No tiene cura. Cada síntoma se trata con un tratamiento distinto.
Galactosemia	<ul style="list-style-type: none">• Se trata de un trastorno hereditario.• Ausencia o discapacidad de las enzimas en convertir la galactosa en glucosa.



	<ul style="list-style-type: none">• El azúcar se acumula en el organismo produciendo problemas de hígado y en el sistema nervioso central [56].• Se diagnostica por medio de una prueba de sangre u orina.• El individuo presenta diarrea, vómito, retraso en el crecimiento, letargia, ...• Si la galactosemia no es diagnosticada y tratada a tiempo puede ocasionar insuficiencia hepática, discapacidad intelectual, retraso en el crecimiento y, en ocasiones, la muerte.• Aun habiendo sido diagnosticado y tratado a tiempo, este trastorno puede desembocar en complicaciones crónicas como el retraso en el crecimiento, dificultades en el habla, problemas motores, dificultades en el aprendizaje, ...• El tratamiento consiste en la restricción total de alimentos y medicamentos con lactosa.
Síndrome alcohólico fetal	<ul style="list-style-type: none">• Es debido a la consumición de alcohol por parte de la madre durante el embarazo.• Provoca que el desarrollo del feto no sea correcto, sobre todo a nivel cerebral.• Los síntomas característicos de este trastorno con los problemas del sistema nervioso central, problemas de crecimiento y características faciales anormales.• El único tratamiento posible que existe para evitar el síndrome alcohólico fetal es la prevención del consumo de alcohol durante el embarazo.

Tabla 2.7. Principales trastornos asociados a la discapacidad intelectual.

2.6.4 Intensidad de los apoyos

Como se ha detallado en la sección 2.6.1, el nivel o intensidad de apoyo que deben recibir las personas con discapacidad intelectual no siempre es el mismo, depende del grado que se posea. Existen cuatro niveles de apoyo o ayuda al individuo: intermitente, limitado, extenso y generalizado [46].

La ayuda que reciben las persona con discapacidad cognitiva que la necesitan únicamente para llevar a cabo actividades o tareas puntuales, se denomina intermitente.

El apoyo de duración limitada (por ejemplo, el adiestramiento laboral) recibido por la persona se denomina ayuda limitada.



Determinados grados de discapacidad intelectual requieren un apoyo que es regular a lo largo de la semana. Se trata de la ayuda extensa. Por ejemplo, realizar las tareas del hogar todos los días.

Finalmente, aquellas personas que poseen una discapacidad cognitiva muy elevada, en ocasiones, necesitan un apoyo de forma casi constante y puede ser de por vida. Se denomina ayuda generalizada.

2.7 Análisis del mercado

Como se ha detallado en la sección 1.1. del documento, actualmente existen aplicaciones orientadas a la ayuda de personas con discapacidad intelectual, pero, sin embargo, profesionales han detectado que los programadores no poseen los suficientes conocimientos del asunto como para llevar a cabo el desarrollo de una aplicación y que en ocasiones las aplicaciones son carácter demasiado general. Para llevar a cabo el desarrollo de la aplicación que se presenta, además de contar con el asesoramiento de un profesional en la educación especializada para escoger correctamente el contenido de los ejercicios de la aplicación, se ha contado con el apoyo del plan docente que imparte la Fundación Prodis a sus alumnos en el programa para personas con discapacidad intelectual Avanzas, prestado por ellos mismos.

A continuación, se ha realizado una búsqueda de las aplicaciones que los usuarios pueden descargar desde sus dispositivos móviles para dar respuesta a las necesidades educativas especiales de las personas con discapacidad intelectual. A pesar de no haber detallado todas las aplicaciones encontradas, se ha podido observar que la gran mayoría de las aplicaciones que actualmente existen en Play Store orientadas a la discapacidad intelectual, se centran en la mayoría de los casos en el autismo, es decir, se centran demasiado en el cuadro clínico explicado en la sección 2.6.3 y no tanto en el sistema de clasificación que se define en DSM V detallado en la sección 2.6.1 (leve, moderado, grave y profundo). Por otra parte, tienen en ocasiones un aspecto demasiado infantil. Como se puede ver en el capítulo 5 del documento, por ejemplo, los chicos del programa Avanzas tienen entre 17 y 18 años y han probado aplicaciones orientadas a ellos y en casos les ha resultado demasiado sencillas o infantiles. Además, todos ellos no padecen autismo. De hecho, el autismo no está considerado como discapacidad intelectual, se trata de un trastorno independiente denominado Trastorno del Espectro Autista que afecta al aprendizaje [65]. Algunas de las aplicaciones encontradas han sido:

- **Terapia del Lenguaje y cognitiva con MITA:** se trata de una aplicación gratuita desarrollada por ImagiRation LLC orientada a niños con autismo, déficit de desarrollo y dificultades en el aprendizaje. En ella, se tratan áreas como el lenguaje, la atención y las habilidades visuales. Además, cuenta con versiones traducidas a portugués, español, francés, italiano, ruso, español y alemán. Actualmente cuenta con más de 100.000 descargas [60].



- **PictoTEA:** es una aplicación gratuita desarrollada por VelociTeam orientada a chicos con Trastornos del Espectro Autismo (TEA), déficit de desarrollo o cualquiera de las afecciones que generan problemas en las habilidades sociales y de la comunicación de la persona. Mediante pictogramas el usuario puede mejorar aspectos de la comunicación con su entorno. Además, se pueden agregar pictogramas propios. La aplicación tiene versiones traducidas al alemán, francés, portugués e italiano. Cuenta con más de 10.000 descargas actualmente [61].
- **iSECUENCIAS LITE:** se trata de una aplicación gratuita desarrollada por la fundación Lovaas orientada a los niños con autismo. Además, en la descripción de la aplicación se detalla que la aplicación también puede ser orientada a niños sin dificultades. Se trabajan o entrenan el área de los hábitos de autonomía, las situaciones cotidianas y las emociones. Actualmente tiene más de 10.000 descargas en Play Store y solo cuenta con versión en español [62].
- **Conciencia fonológica:** es una aplicación gratuita desarrollada por Miniworld Apps orientada a niños con autismo, déficit de desarrollo y dificultades en el aprendizaje. En dicha aplicación, el niño aprenderá a leer. Por otro lado, en la descripción de la aplicación se detalla que podría ser útil para niños preescolar o educación infantil. Cuenta con más de 50.000 descargas y solamente tiene versión en español [63].
- **Soy Cappaz:** se trata de una aplicación gratuita desarrollada por la Fundación Mapfre que actualmente cuenta con más de 10.000 descargas. Soy Cappaz está orientada a personas que tienen capacidades distintas ayudando a que posean una vida más autónoma. Cuenta con una versión en portugués [64].

En definitiva, la mayoría son orientadas principalmente al autismo y tienen un aspecto infantil como para ser utilizada por chicos de aproximadamente 18 años. Es por ello por lo que se ha desarrollado una aplicación que se ciñe a los ejercicios que las personas con discapacidad intelectual leve están acostumbradas a hacer con el fin de la estimulación cognitiva. Además, se ha intentado que su aspecto no sea demasiado infantil.

CAPÍTULO 3. Descripción general del sistema

En este capítulo de la memoria del Trabajo Fin de Grado, en primer lugar, se ha explicado el funcionamiento general y las distintas actividades que se pueden realizar en la aplicación desarrollada.

Posteriormente, se menciona la versión de Android que se ha utilizado durante la programación de la aplicación al mismo tiempo que se explica de forma breve cómo instalar el entorno de desarrollo utilizado, Android Studio.

Además, se expone la estructura que tiene el proyecto Android entrando en detalle en los subdirectorios que la componen.

También se mencionan y analizan los principales elementos y componentes de la programación Android utilizados para llevar a cabo el desarrollo de la aplicación.

Finalmente, se explican los recursos que hacen posible el correcto funcionamiento de la aplicación.

3.1 Descripción de la aplicación

Se ha llevado a cabo el desarrollo de una aplicación para dispositivos Android la cual tiene como principal objetivo dar respuesta a las necesidades educativas especiales de las personas con discapacidad intelectual leve. Además, en todo momento se ha pensado que la interfaz de la aplicación debía ser lo más intuitiva posible para que los chicos y chicas que padecen este tipo de discapacidad además de poder utilizarla durante las clases de los programas de educación especial, pudiesen utilizarla en sus respectivas casas sin ayuda de un responsable.

En el momento en el que el usuario abre la aplicación se encuentra el menú principal con cinco botones, donde cada uno de ellos está relacionado con un área de trabajo: lengua, matemáticas, cultura, inglés y autogestión.

En la sección de lengua se trabaja la ordenación de palabras para formar frases, la comprensión lectora y la memoria.

En la sección de matemáticas se realiza un entrenamiento de operaciones con números enteros, de fracciones y del euro.

En el botón de cultura, el usuario puede jugar y entrenar sus conocimientos culturales a través de un juego de preguntas tipo test.

Mediante un juego de memoria el usuario podrá practicar las palabras más básicas del idioma inglés.

Finalmente, en la sección de autonomía, el individuo puede practicar con las horas, la ordenación de frases para llevar a cabo una actividad (secuencia) y la toma de decisiones.



A continuación, en la Figura 3.1., se muestra un diagrama de las actividades que contiene la aplicación desarrollada para el Trabajo Fin de Grado.

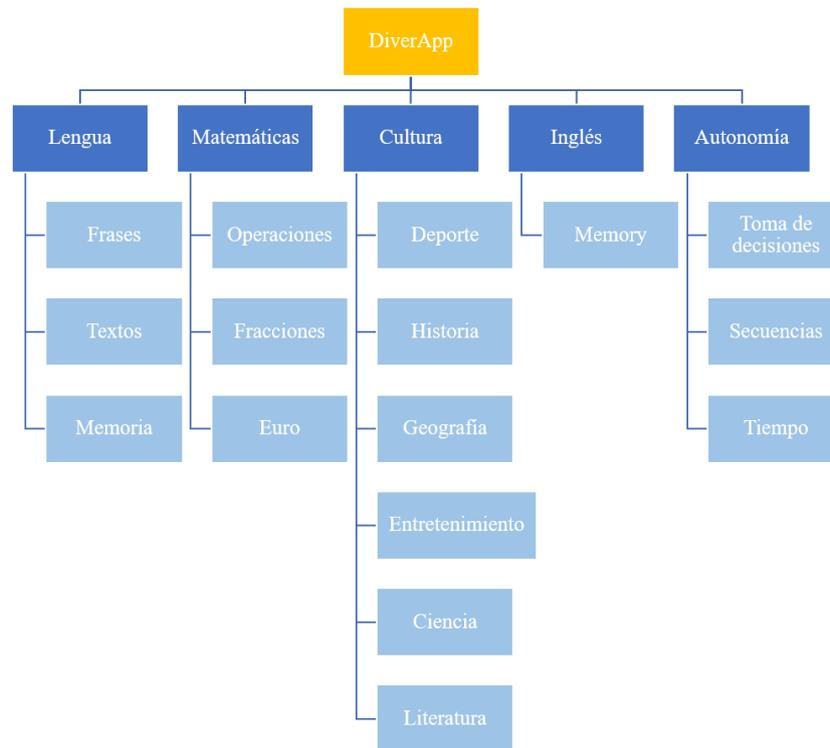


Figura 3.1. Estructura de la aplicación desarrollada

3.2 Herramientas del sistema

3.2.1 Versión del SDK de compilación

Para llevar a cabo el desarrollo de la aplicación, se ha utilizado la versión 27 de SDK, es decir, la versión Oreo de Android (8.1.). Como se puede ver en la Tabla 2.1. y en el gráfico de la Figura 2.5 de la sección 2.1.2. del documento, Oreo es la versión más distribuida en octubre de 2018 entre los dispositivos móviles Android con un 28%.

No se conoce si la versión de Android Pie ha sufrido un crecimiento en cuanto a su distribución ya que Android no ha publicado la distribución de versiones correspondientes al año 2019. Es por ello por lo que se ha preferido utilizar una versión que recientemente ha sido y es actualmente la versión más empleada por los usuarios.

3.2.2 Instalación de Android Studio:

En este apartado se procede a explicar los pasos a seguir para instalar el entorno de desarrollo escogido y explicado en la sección 2.2., Android Studio.

En primer lugar, se ha de acceder a la web [66] y descargar el archivo *.exe* recomendado por Android. En este caso se ha de descargar el marcado en color rojo en la Figura 3.2.

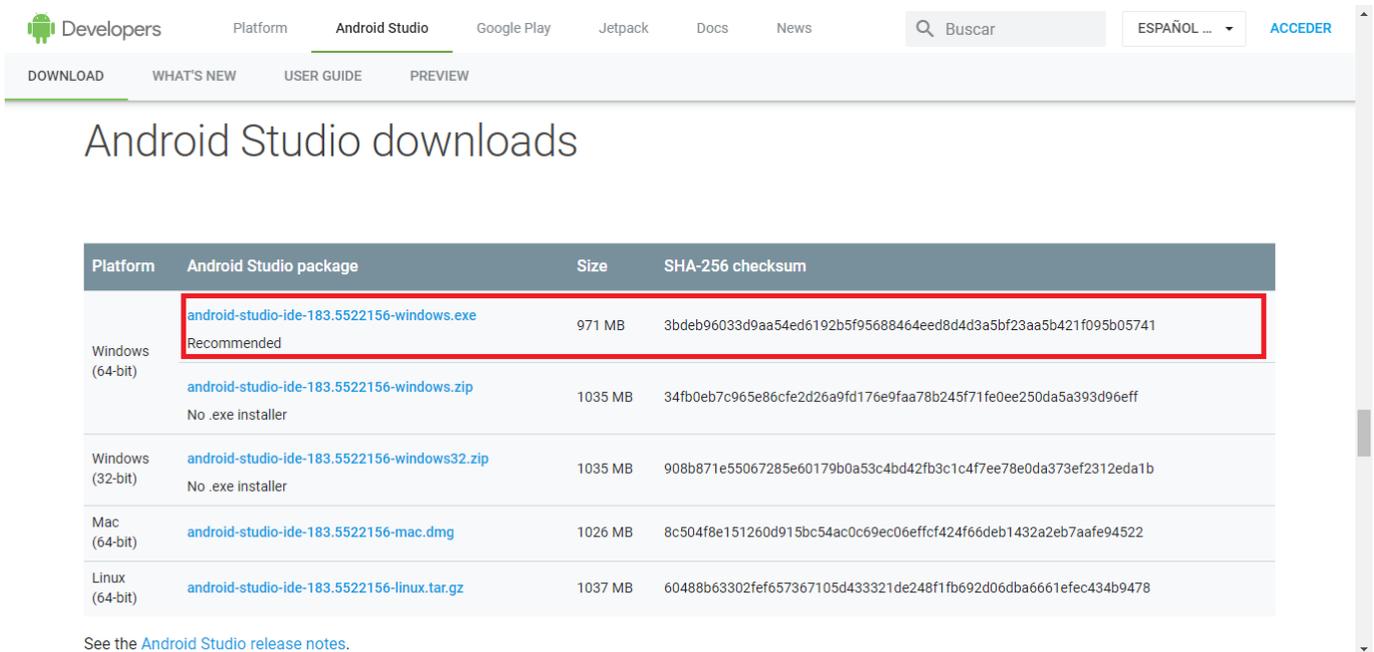


Figura 3.2. Descarga del archivo ejecutable de Android Studio.

Previamente, como se puede observar en la Figura 3.3., se han de leer y aceptar los términos y condiciones de Android Studio.

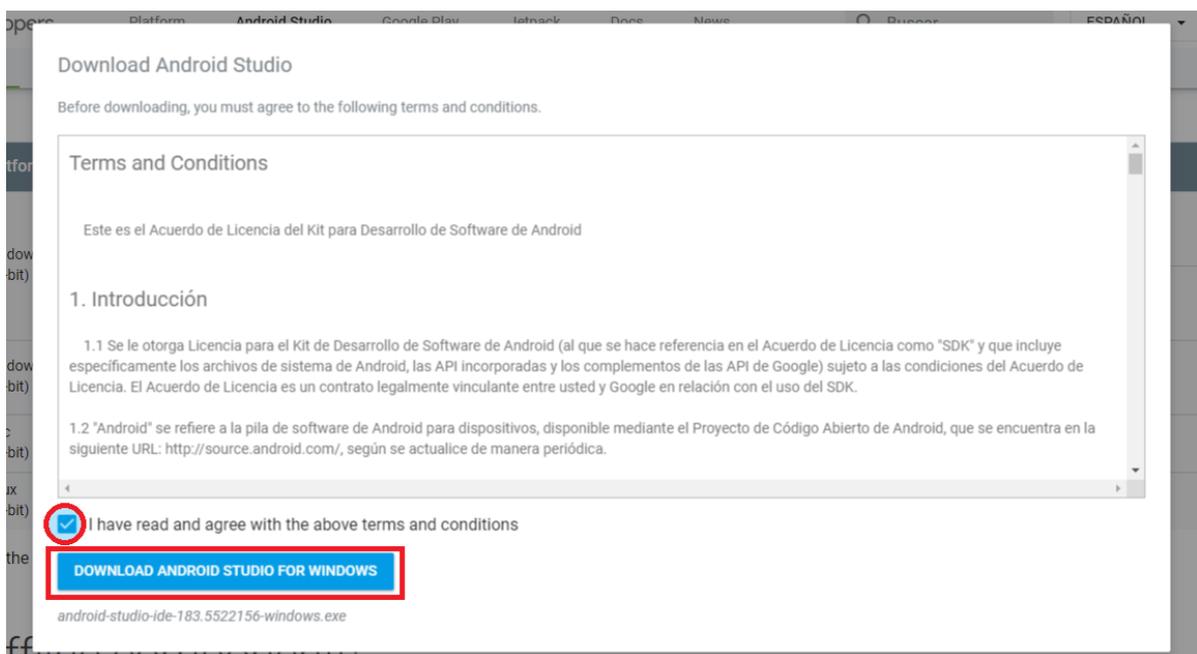


Figura 3.3. Términos y condiciones previos a la descarga de Android Studio.

Una vez finalizada la descarga del archivo, se ejecuta y aparecerá una ventana donde Android Studio avisa de que está verificando el archivo instalador para posteriormente abrir un asistente de instalación que se muestra en la Figura 3.4:

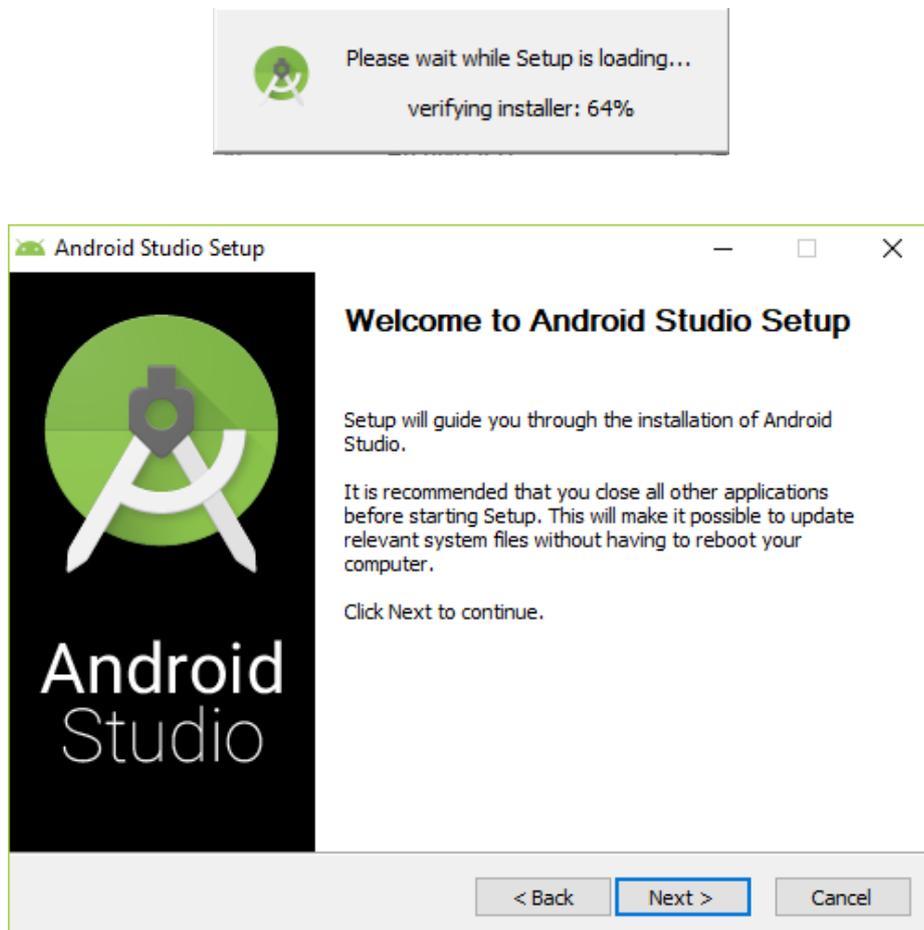


Figura 3.4. Asistente de instalación

Tras hacer clic en el botón ‘Siguiente’, el instalador trata de encontrar y configurar el entorno de desarrollo java (JDK). En el caso de que no lo encuentre aparece una ventana para que el usuario especifique su localización. En este caso, como se especifica en la sección 2.2.3., se ha descargado el JDK 8 para Windows mediante el cual se pueden desarrollar aplicaciones con el uso del lenguaje Java en una plataforma Java.

Posteriormente, en la siguiente ventana, el usuario debe seleccionar las componentes que desea instalar a parte del propio programa de Android Studio. El asistente de la instalación da las opciones de el Android SDK y de *Android Virtual Device* (AVC). Para desarrollar la aplicación del Trabajo Fin de Grado, se han descargado ambas componentes a pesar de que AVC no haya sido muy utilizado ya que las simulaciones se han realizado en dispositivos móviles reales.

En la Figura 3.5. no aparece la descarga de la componente SDK porque ya está instalada de la versión anterior de Android Studio.

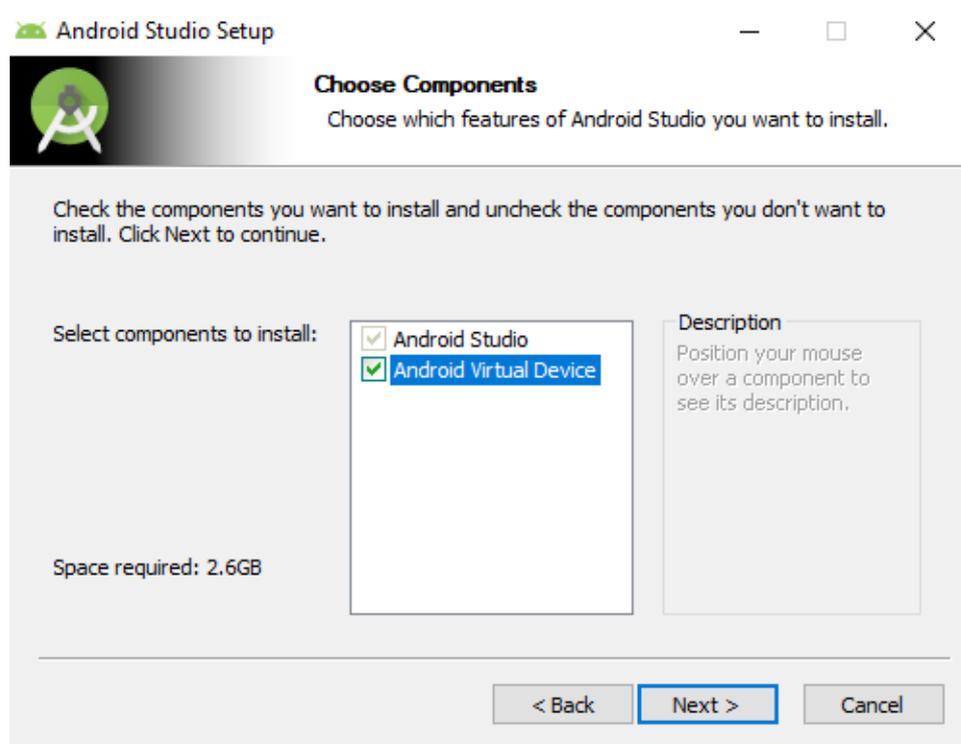


Figura 3.5. Descarga de componentes

Una vez pulsado el botón 'Next', aparece una ventana donde se debe especificar la localización que tiene el propio programa de Android Studio y la localización del SDK de Android, tal y como se puede observar en la Figura 3.6.

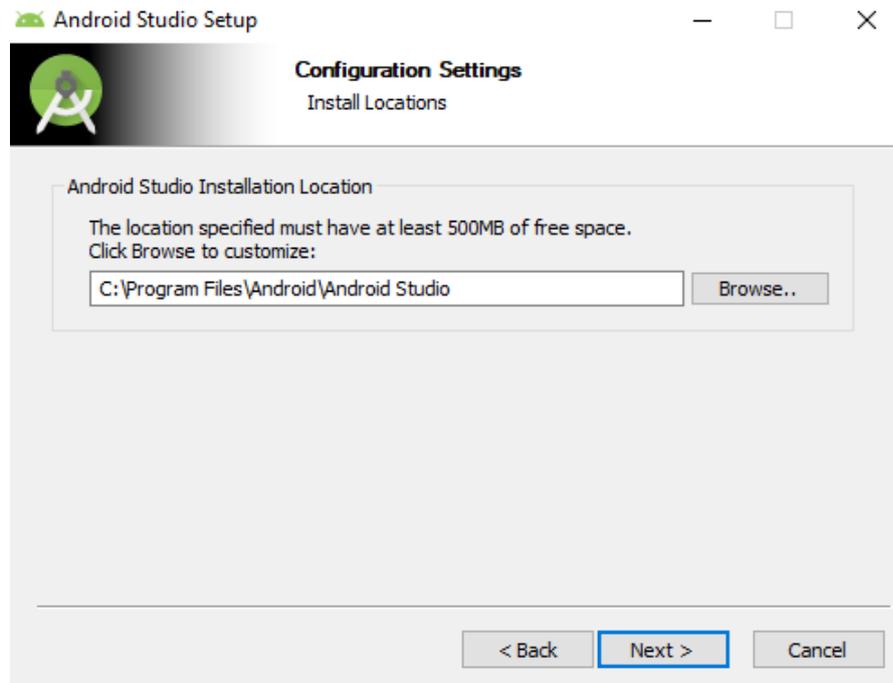


Figura 3.6. Localización de Android Studio y el SDK de Android.

Antes de comenzar la instalación, el asistente pregunta si se desea un acceso directo desde el escritorio y tras ello, la instalación de Android Studio empieza cuando se pulse el botón 'Install'. En la Figura 3.7 se muestra este último paso.

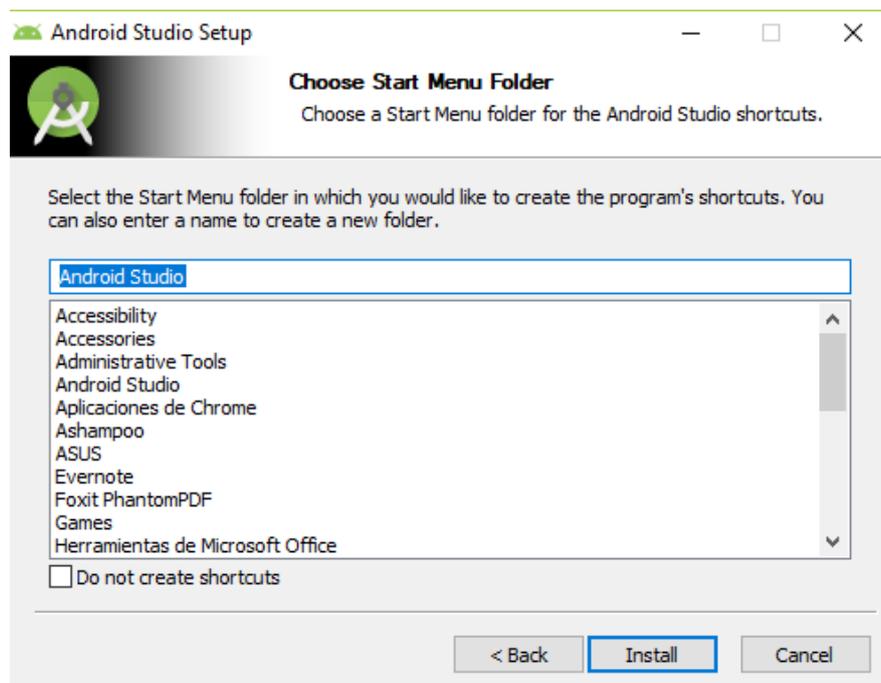


Figura 3.7. Instalación de Android Studio.

Una vez que ha concluido la instalación de Android Studio, el desarrollador ya está listo para comenzar a programar su aplicación desde la pantalla de la Figura 3.8.

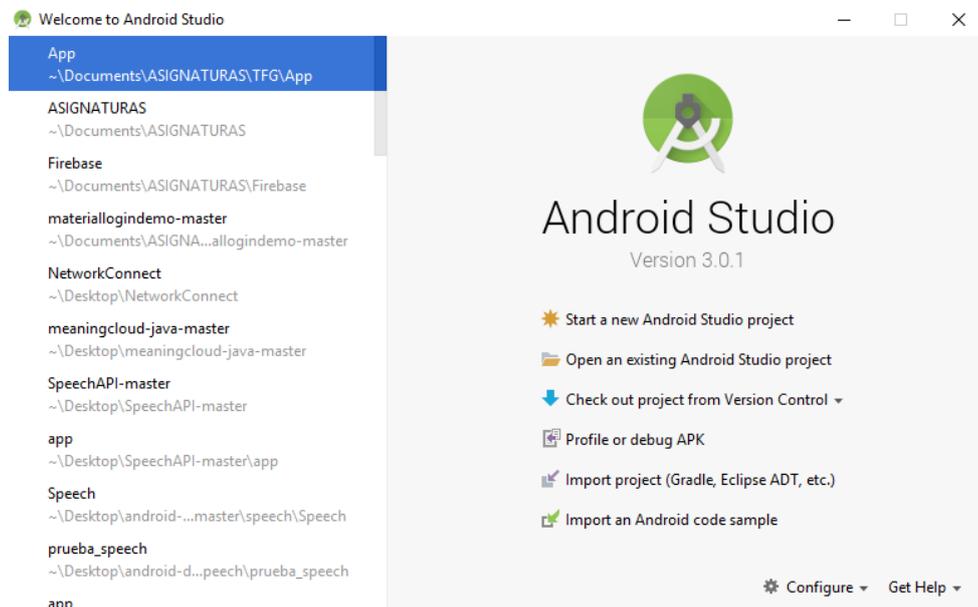


Figura 3.8. Menú de inicio de Android Studio

3.2.3 Instalación de herramientas en SDK Manager

Cuando se lleva a cabo la instalación de Android Studio, se descargan de forma automática algunas de las herramientas que ofrece Android por defecto. Sin embargo, como ya se ha explicado en la sección 3.2.1 del documento, para el desarrollo de la aplicación se ha utilizado la versión 8.1 de Android que se corresponde con el API 27. Por ello, se ha procedido a descargar los paquetes correspondientes a 'Android API 27' que se señala en la Figura 3.9.

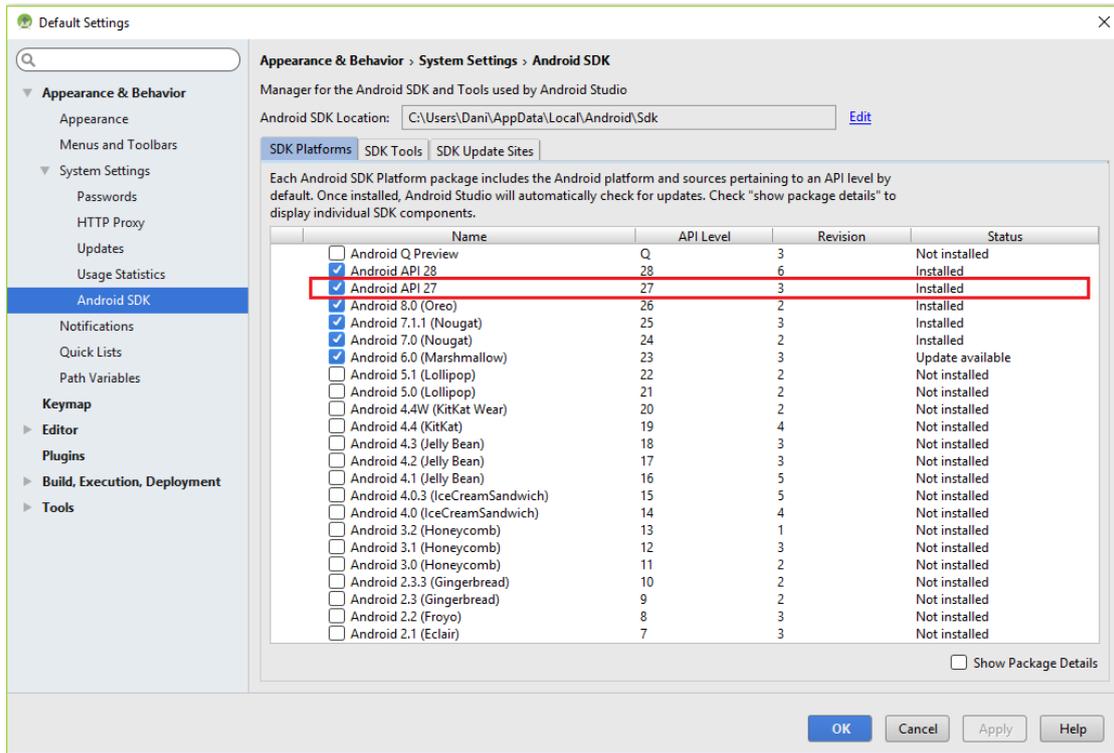


Figura 3.9. Descarga de los paquetes correspondientes a Android API 27 (Oreo 8.1).

Por otra parte, no ha hecho falta descargar ningún paquete adicional desde la pestaña de herramientas SDK (SDK Tools). Únicamente ha habido que actualizar la versión de la herramienta Android SDK Tools.

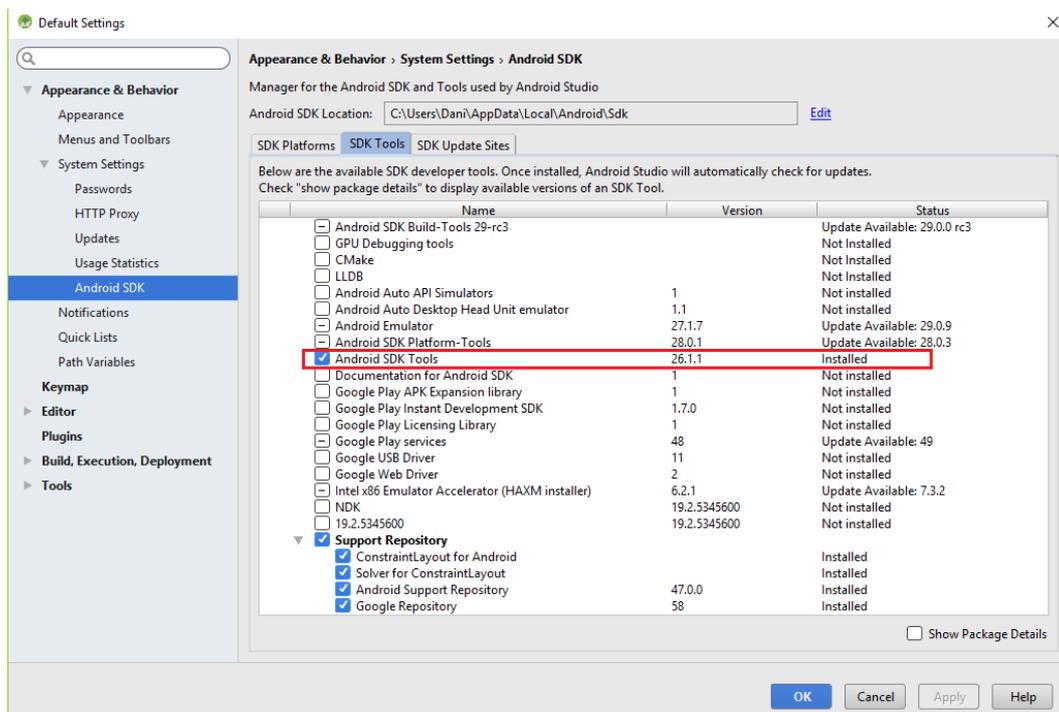
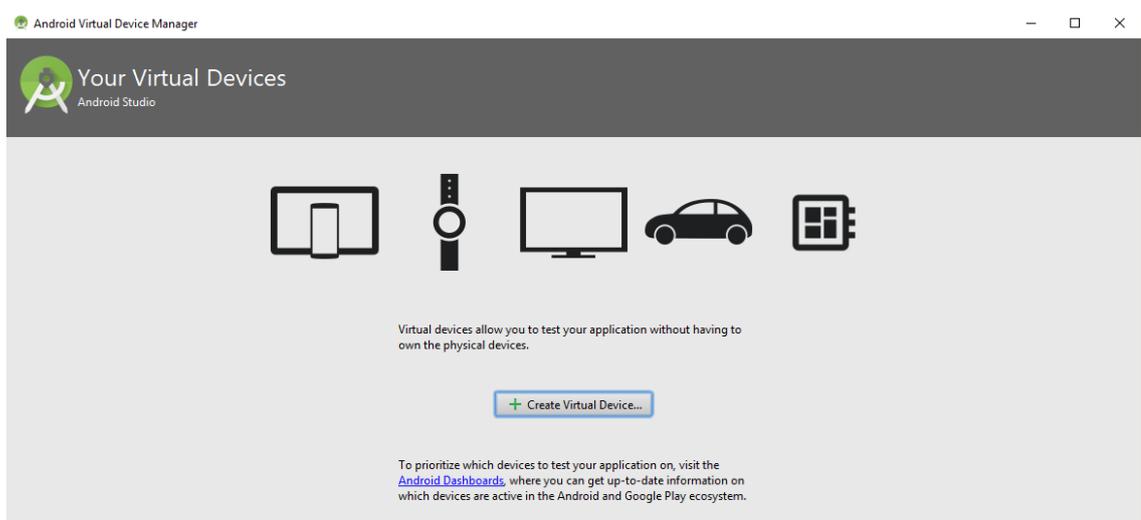
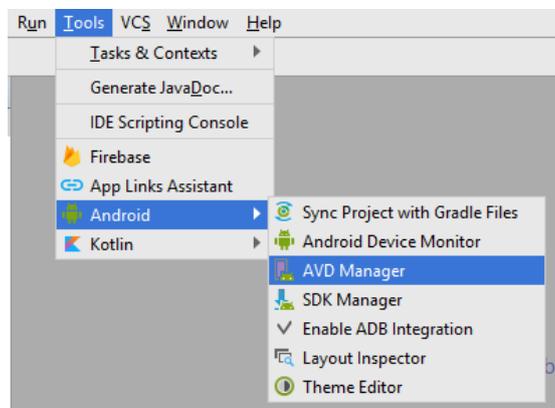


Figura 3.10. SDK Tools.

3.2.4 Configuración AVD vs. Dispositivos móviles reales:

Como ya se ha comentado en puntos anteriores, las simulaciones de la aplicación desarrollada se han llevado a cabo en dispositivos móviles reales en vez de utilizar la herramienta AVD que ofrece Android Studio.

En cualquier caso, para configurar los dispositivos móviles virtuales que se utilizan con AVD, se debe acceder a AVD Manager desde *Tools > Android > AVD Manager*. Posteriormente, el usuario puede crear el dispositivo móvil virtual de forma personalizada pudiendo elegir tipo de dispositivo, tamaño del dispositivo, su resolución y su densidad de píxeles, tal y como se muestra en la Figura 3.11.



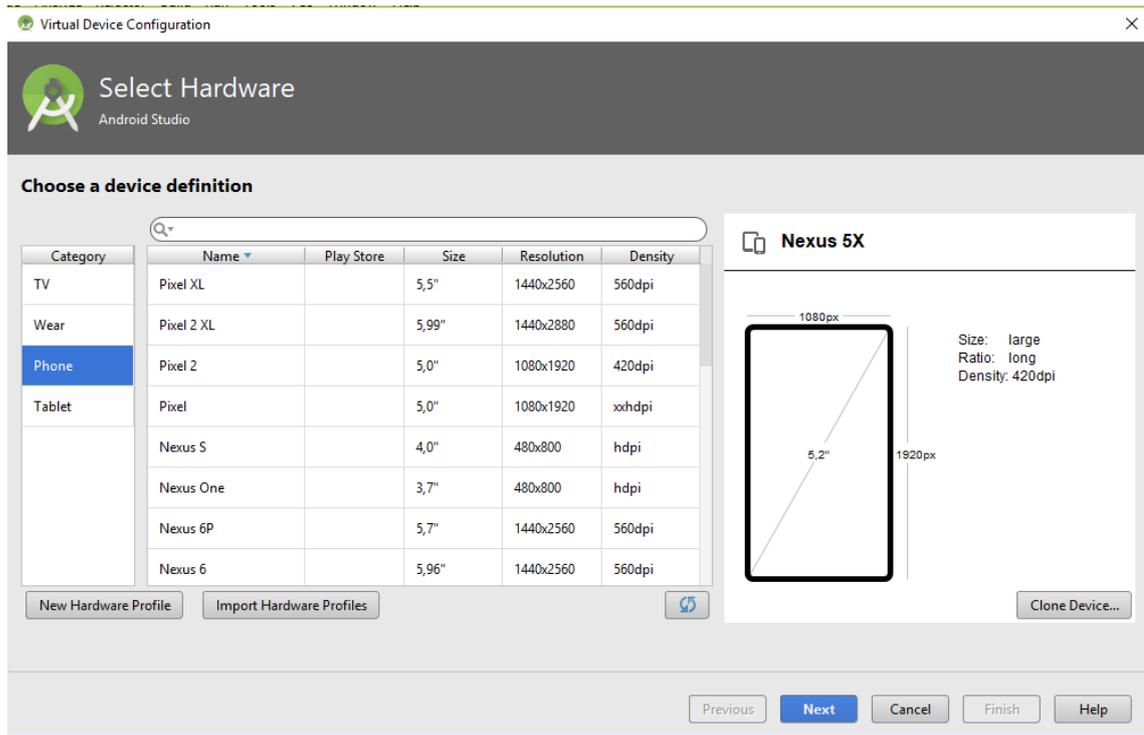


Figura 3.11. Crear dispositivo móvil virtual en AVD Manager.

Se han utilizado 4 dispositivos móviles reales distintos para la simulación de la aplicación desarrollada. El primero de ellos tiene versión Android 6.0 (Marshmallow), la cual se corresponde con el nivel de API 23. El segundo dispositivo cuenta con la versión Android 6.0.1 (Marshmallow), que también se corresponde con el nivel de API 23. Por otro lado, el tercer dispositivo utilizado tiene la versión Android 7.0 (Nougat) y por tanto su nivel de API es el 24. Finalmente, el cuarto y último dispositivo utilizado para realizar las simulaciones tiene versión 8.0.0 (Oreo), que se corresponde con el nivel de API 26.

Por tanto, se ha podido comprobar que a pesar de haber diseñado la aplicación con el paquete SDK de Android API 27, los dispositivos de sistema operativo Android con niveles de API desde la 23, funcionan correctamente. Además, cada uno de los dispositivos móviles es de tamaño distinto y se ha podido ver que el diseño de la aplicación se amolda correctamente independientemente del tamaño.

3.3 Sistema de ficheros

Tal y como se detalla en la sección 2.2.2. del documento, los proyectos Android pueden tener uno o varios módulos que compongan la aplicación en cuestión. En este caso, únicamente se ha desarrollado un módulo el cual cuenta con los directorios: *manifests*, *java*, *res* y *Gradle Scripts*, como se puede ver en la Figura 3.12.

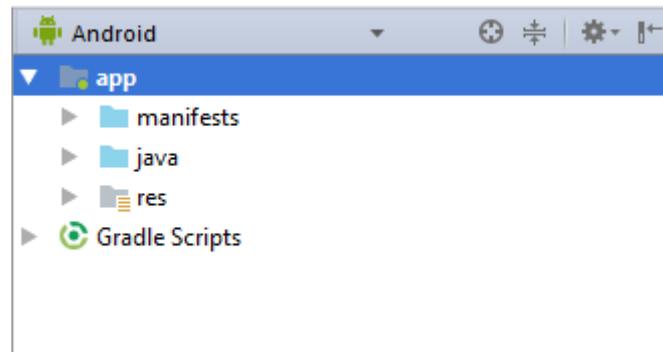


Figura 3.12. Sistema de ficheros del proyecto

En el directorio *manifests* se encuentra el manifiesto, como se observa en la Figura 3.13., aquel archivo que deben poseer todas las aplicaciones donde se almacena la información necesaria para poder ejecutar el código. Se recuerda que la información detallada del archivo manifiesto se encuentra en la sección 2.1.5.

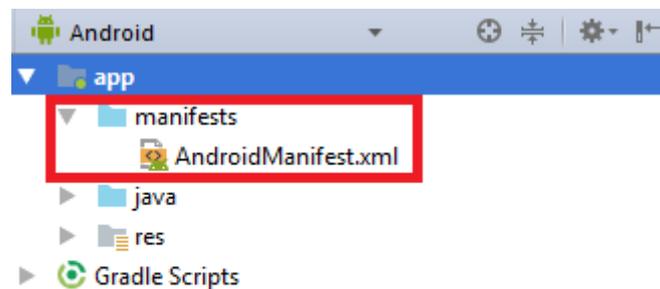


Figura 3.13. Directorio *Manifests*

Por otro lado, en el directorio *java*, tal y como se especifica en la sección 2.2.2., se han almacenado todos los archivos *.java* en el interior del paquete de la aplicación. Dicho subdirectorio se muestra en la Figura 3.14.

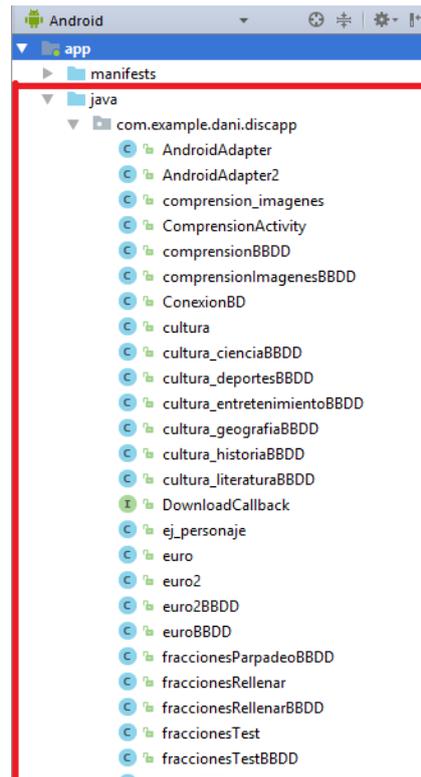


Figura 3.14. Directorio java

En el directorio *res*, se pueden encontrar los ocho subdirectorios que aparecen en la Figura 3.15.:

- *Anim*: es el lugar donde se almacenan todos los archivos *.xml* correspondientes a las animaciones que se han utilizado durante el desarrollo de la aplicación.
- *Drawable*: en él se encuentran todas las imágenes que aparecen en la aplicación además de los archivos *.xml* correspondientes a los *shapes* o formas utilizados en algunos botones.
- *Font*: en este subdirectorio se han guardado las fuentes de texto que se utilizan en la aplicación.
- *Layout*: es el lugar donde se encuentran todos los archivos *.xml* correspondientes a los layouts que conforman la aplicación.
- *Menu*: en él se encuentra el archivo *.xml* correspondiente al layout del *Navigation Drawer* que contiene la aplicación desarrollada.
- *Mipmap*: mediante archivos *.xml* se define el diseño del icono de la aplicación además de la imagen que lo compone en los formatos de densidad (hdpi, mdpi, xhdpi, xxhdpi y xxxhdpi).

- *Raw*: es el subdirectorio donde se han almacenado todos los archivos *.txt* que inicializan la base de datos SQLite y todos los archivos de audio que se utilizan en la aplicación.
- *Values*: en él se puede encontrar los archivos *.xml* donde se definen los colores que se han utilizado en el diseño de la aplicación (*color.xml*), las dimensiones (*dimens.xml*), los textos (*strings.xml*) y los estilos (*styles.xml*).

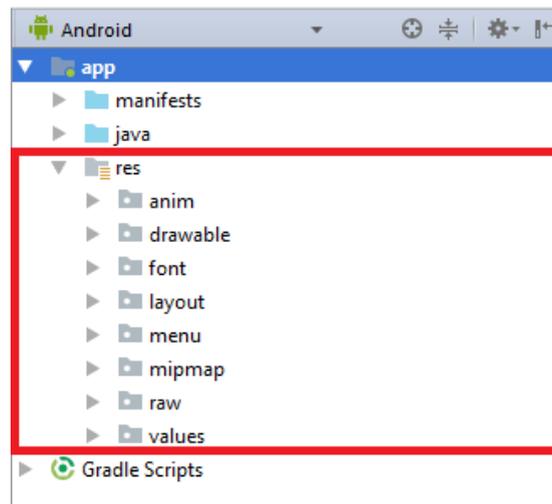


Figura 3.15. Directorio *res*.

Finalmente, en *Gradle Scripts* el usuario puede reconocer los dos archivos configurables de la compilación Gradle que se mencionan en la sección 2.1.6. de la memoria: *build.gradle* (del proyecto y del único módulo de la aplicación) y *settings.gradle*. Dichos ficheros se muestran en la Figura 3.16.

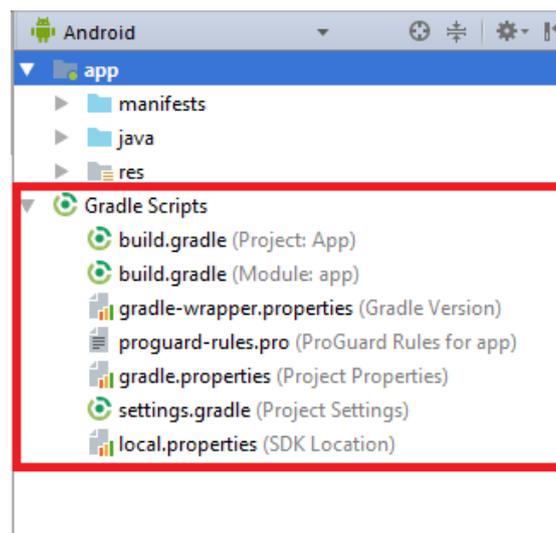


Figura 3.16. Directorio *Gradle Scripts*

3.4 Principales componentes Android utilizados:

El framework de Android está formado por siete componentes que son los que conforman una aplicación. En esta sección de la memoria se abordan los principales componentes Android que han sido utilizados para completar el desarrollo de la aplicación [6].

3.4.1 Activity o actividad

Son las pantallas con las que los usuarios pueden interactuar con el objetivo de llevar a cabo una determinada acción. La aplicación desarrollada cuenta con un gran número de actividades. Además, algunas de las actividades de la aplicación son las encargadas de iniciar otras.

Sin embargo, cada vez que se inicia una actividad nueva ésta se almacena un por encima de la anterior en la pila de actividades, que se detiene. La actividad que esté en un nivel más superior que el resto de las actividades, es la actividad que se muestra en pantalla. Se puede observar el proceso en la Figura 3.17, donde la actividad 1 es la actividad anterior a la actividad 2.

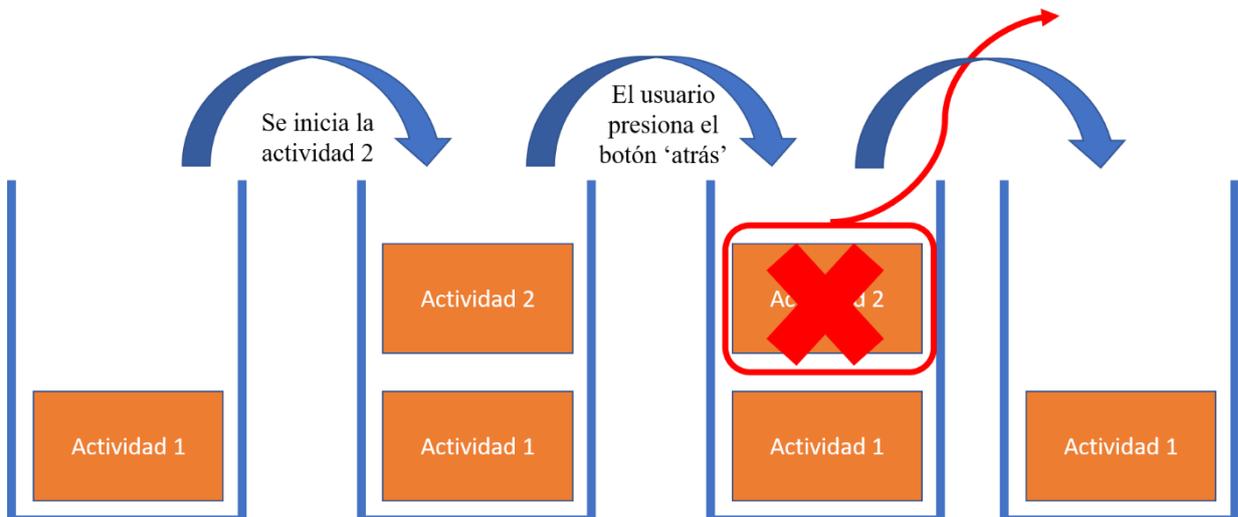


Figura 3.17. Pila de actividades

Por tanto, en el desarrollo de la aplicación se ha implementado en todas las actividades el método *callBack()*. A cada actividad se le ha asignado una segunda actividad en el caso de que el usuario quiera retroceder y, de esta manera, se evitan comportamientos no deseables por parte del programa como que retroceda a la actividad anterior que hay almacenada en la pila cuando se desea acceder a otra distinta. Es por ello por lo que se ha de tener en cuenta y entender bien el ciclo de vida de una actividad Android que se muestra en la Figura 3.18:

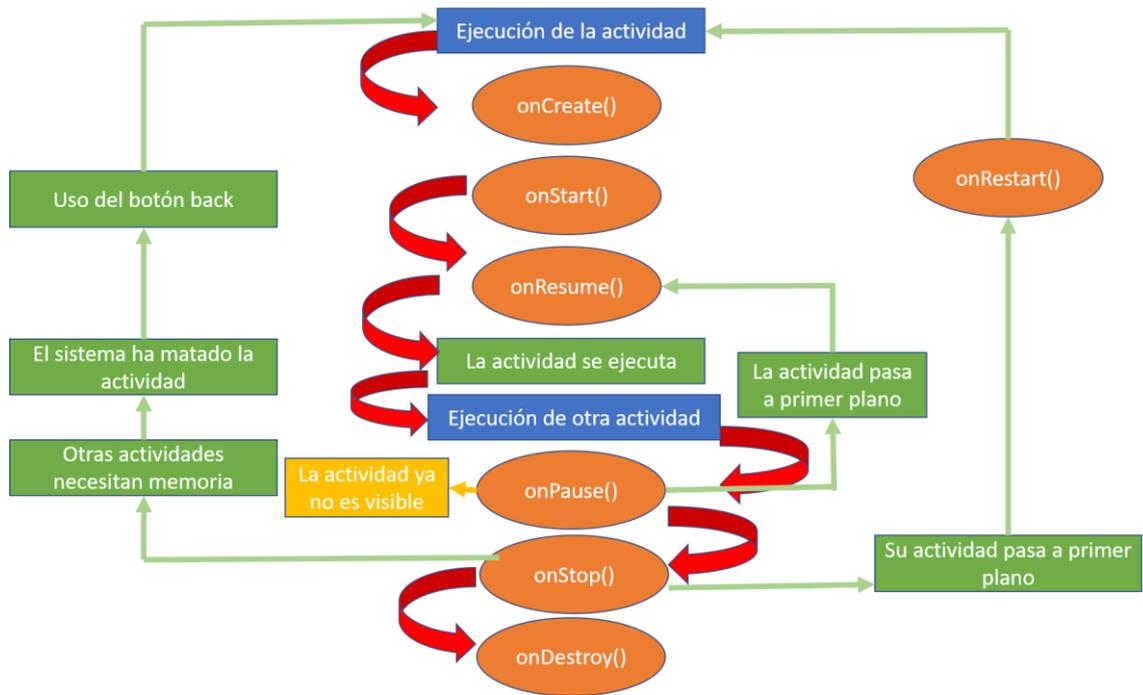


Figura 3.18. Ciclo de vida de una actividad.

3.4.2 Fragmento

Un fragmento se puede definir como parte de una interfaz o actividad. De esta manera, una actividad puede estar compuesta por una o varios fragmentos, los cuales propician que dicha interfaz sea más dinámica y flexible. En la Figura 3.19 se observa la composición de una actividad a través de dos fragmentos. Sin embargo, en la Figura 3.20 se muestra como una actividad puede estar compuesta por un único fragmento

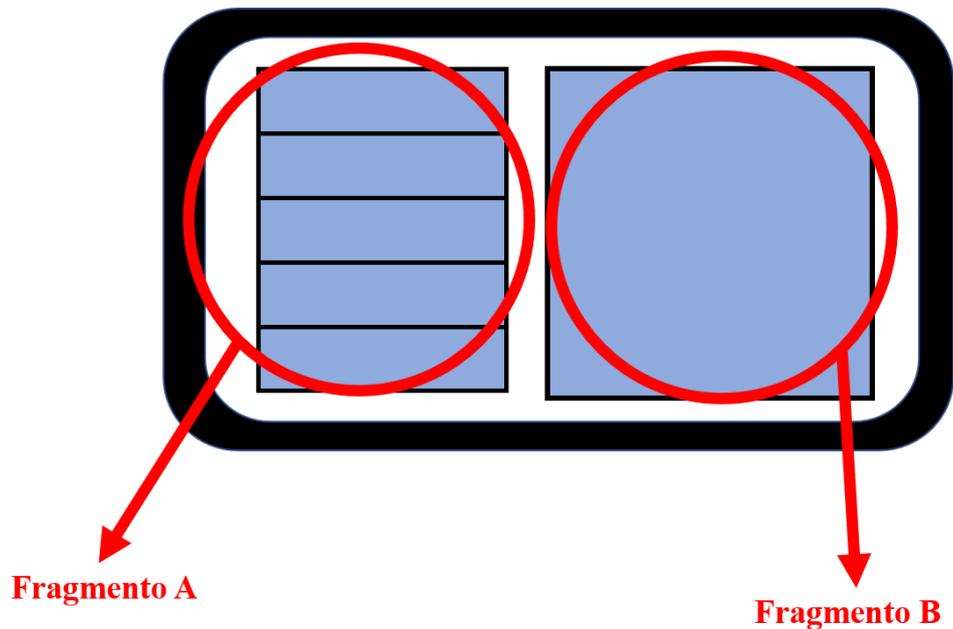


Figura 3.19. Dos fragmentos componen una actividad.

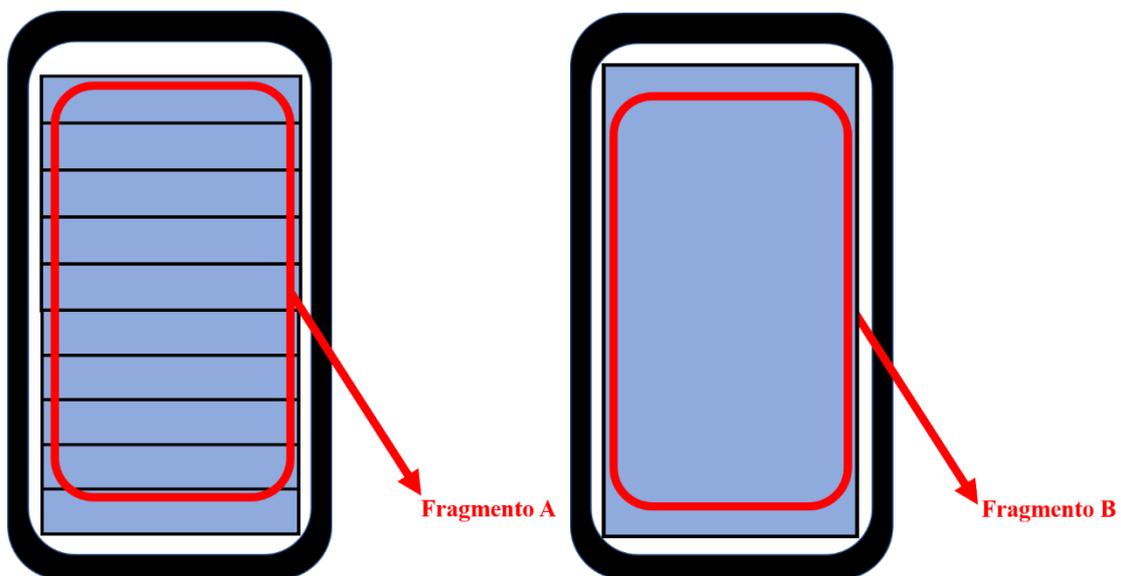


Figura 3.20. Dos actividades distintas cada una con un fragmento distinto.

Los fragmentos de una actividad no siempre hacen referencia a elementos de la interfaz de usuario (IU). En ocasiones, ejecutan por ejemplo acciones de red en

segundo plano, como es el caso de la clase NetworkFragment, la cual extiende de Fragment. En la aplicación desarrollada para el proyecto fin de grado, se ha hecho uso de esta clase para realizar las tareas necesarias para conectar con el servicio de MeaningCloud desde el menú principal.

3.4.3 Intents

Son los mensajes que emiten los componentes de una aplicación Android para comunicarse con el resto de las aplicaciones cuando se produce un evento, por ejemplo, iniciar la aplicación del teléfono para realizar una llamada. Sin embargo, en este caso, se han utilizado los intents para iniciar las distintas actividades, algo que también es posible.

Por tanto, se pueden distinguir dos tipos distintos de intents: explícitos e implícitos. El intent implícito se utiliza con el objetivo de que el sistema ejecute alguna acción específica como es abrir la aplicación de la cámara de fotos para tomar una. Por otro lado, el intent explícito sirve para iniciar nuevas actividades.

- Intent-filters o filtros de intención: son utilizados para definir los tipos de intents, los datos y las categorías de intents que acepta una actividad. Cada filtro se debe desarrollar en el archivo manifiesto del proyecto. En este caso se ha utilizado el siguiente filtro en el cual se indica que la actividad ‘inicio’ es la actividad principal y que estará disponible su lanzamiento desde el menú de inicio de aplicaciones de Android. La Figura 3.21 muestra una captura de pantalla del intent-filter utilizado en la actividad de nombre ‘inicio’ de la aplicación desarrollada.

```
<activity
  android:name=".inicio"
  android:theme="@style/AppTheme.NoActionBar">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

Figura 3.21. Intent-filter de la actividad ‘inicio’

3.4.4 Clase Application

- Toda aplicación Android cuenta con la clase Application, la cual permanece instanciada durante el ciclo de vida de la aplicación. Es utilizada para la declaración de constantes globales de la aplicación que se desarrolla, proteger las variables globales que se utilizan en la aplicación y encargarse de gestionar los recursos utilizados por la aplicación. Dicha clase se debe declarar al inicio del manifiesto del proyecto, de esta manera, será lo primero que se instancie en el ciclo de vida de la aplicación. Posteriormente, tal y como se detalla en la

sección 2.1.5 del documento, en su interior, será el lugar donde se declaren todas las actividades que compongan la aplicación que se está desarrollando.

3.5 Interfaces

En esta sección de la memoria se detallan los principales elementos utilizados para la creación de las interfaces que componen la aplicación desarrollada.

3.5.1 Layouts

Se denomina layout a los contenedores de los elementos que componen una vista.

- **FrameLayout:** se trata de un contenedor vacío en el que se puede mostrar cualquier objeto.
- **LinearLayout:** existen dos variantes de este tipo de contenedor: vertical y horizontal. En el LinearLayout vertical los elementos que se encuentran en el interior del contenedor, se colocan de manera vertical uno debajo de otro. Por otro lado, en el LinearLayout horizontal, los elementos se colocan uno a la derecha de otro.
- **RelativeLayout:** es el tipo de contenedor en el que un elemento se coloca en función de otro, por ejemplo, la imagen B va a ser colocada a la izquierda de la imagen A. Además, los elementos también pueden ser colocados en función del mismo contenedor, por ejemplo, la imagen A va a ser alineada con el borde derecho del contenedor.
- **ConstraintLayout:** es el contenedor utilizado para definir las restricciones de una vista con respecto al layout padre, a otra vista o definir su alineación con respecto a otra vista. En el caso de la aplicación desarrollada, se ha utilizado para definir las restricciones de algunas vistas con respecto al ancho y alto de la pantalla, es decir, algunos elementos ocupan un tanto por ciento de ancho y un tanto por ciento de alto con respecto al tamaño de la pantalla del dispositivo donde se ejecute la aplicación. Para poder utilizar este tipo de Layout se ha tenido que añadir la siguiente línea de código a las dependencias del módulo:

```
Implementation 'com.android.support.constraint:constraint-  
layout:1.1.3'
```

3.5.2 View o vistas

En la clase View es donde se encuentran cada uno de los elementos que son almacenados en los Layout y son los encargados de llevar a cabo el diseño de la aplicación. A continuación, se van a detallar aquellos elementos que han sido utilizados durante el desarrollo de la aplicación Android.

- **TextView:** es el elemento mediante el cual se le muestran al usuario de la aplicación textos.



- **EditText:** en la aplicación creada se utiliza para uno de los juegos de fracciones, en el que el jugador debe introducir mediante el teclado números.
- **Button:** conforman cada uno de los botones de la aplicación.
- **ImageView:** las imágenes que se muestran en los juegos han sido definidas mediante este tipo de elementos.
- **ViewPager:** se trata de un elemento que cuenta con todas las páginas de visualización que se deseen. Mediante el deslizamiento sobre la región que ocupa el ViewPager, se produce un cambio de página y se visualizan los elementos definidos en esa página.
- **ListView:** es un elemento de Android el cual consiste en mostrar una serie de elementos uno a continuación de otro. Como máximo cada uno de los elementos de una lista puede poseer tres líneas y cada una de las filas puede estar compuesta por los elementos definidos.

3.5.3 Shapes

Los elementos Shape son elementos creados en ficheros *.xml* que han sido almacenados en el subdirectorio *Drawable* y en los que mediante comandos se han definido dibujos de figuras geométricas para ser utilizados como *background* de algunos elementos. Por ejemplo, se ha diseñado el rectángulo que se muestra en la Figura 3.22. para ser el fondo de las respuestas tipo test de algunos juegos de la aplicación.



Figura 3.22. Shape de respuestas tipo test

3.6 Firebase. Autenticación de usuarios

Al igual que en muchos casos es necesario realizar una distinción entre los usuarios que utilizan una aplicación, para el desarrollo de este proyecto también se ha hecho. Mediante la plataforma de desarrollo Firebase se ha introducido un servicio de autenticación de usuarios en el que nada más arrancar la aplicación, el usuario, si aún no está registrado, debe introducir un correo y una contraseña y pulsar el botón ‘Registrar usuario’. Por otro lado, si el usuario ya ha sido registrado en el sistema, simplemente debe introducir el correo y la contraseña con las que lo hizo y pulsar el botón de ‘Login’ para comenzar a hacer uso de la aplicación.

Tal y como se detalla en la sección 2.5.2. del documento, tras haber conectado la aplicación a una cuenta Firebase, haber introducido las dependencias correspondientes y haber sincronizado el proyecto para que las dependencias tengan las versiones correctas, se ha de configurar el proyecto Firebase desde la web [39].

En primer lugar, debemos seleccionar el proyecto Firebase al que está conectada la aplicación. Para realizar una explicación paso a paso, se ha creado un nuevo proyecto ‘AutenticacionTFG’ con el fin de llevar a cabo la configuración desde cero. En la Figura 3.23. se muestra el menú de selección de proyectos Firebase.

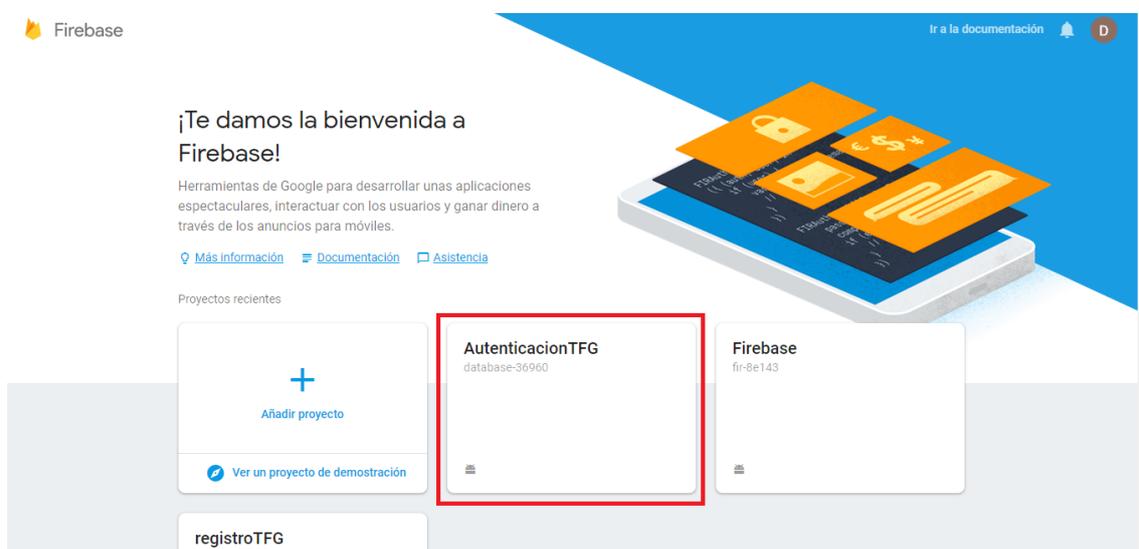


Figura 3.23. Menú de selección de proyecto Firebase

Una vez seleccionado el proyecto que se desea, aparece la pantalla principal de la consola. Como se puede ver en la Figura 3.24., a la izquierda de pantalla aparecen todos los servicios que ofrece Firebase. Para configurar la parte de autenticación de usuarios del proyecto, se debe acceder a ‘Authentication’.

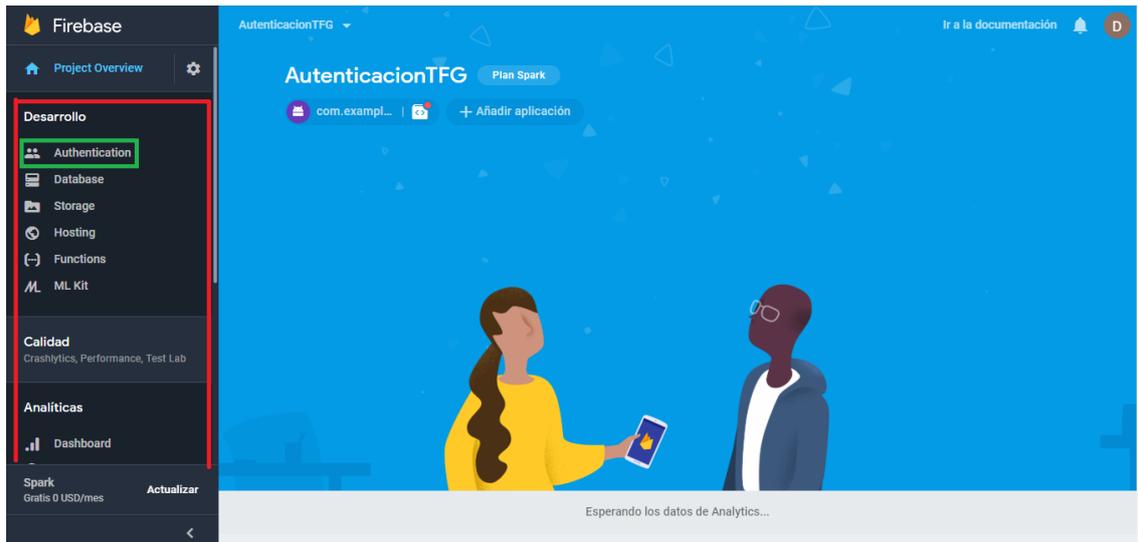


Figura 3.24. Menú de selección de servicio de Firebase (En rojo todos los servicios que ofrece Firebase. En verde el servicio de Autenticación en el que se basa esta sección)

Dentro del apartado de autenticación de la consola de Firebase, aparece la pantalla que se muestra en la Figura 3.25. En ella se debe hacer clic en ‘Configura el método de inicio de sesión’.

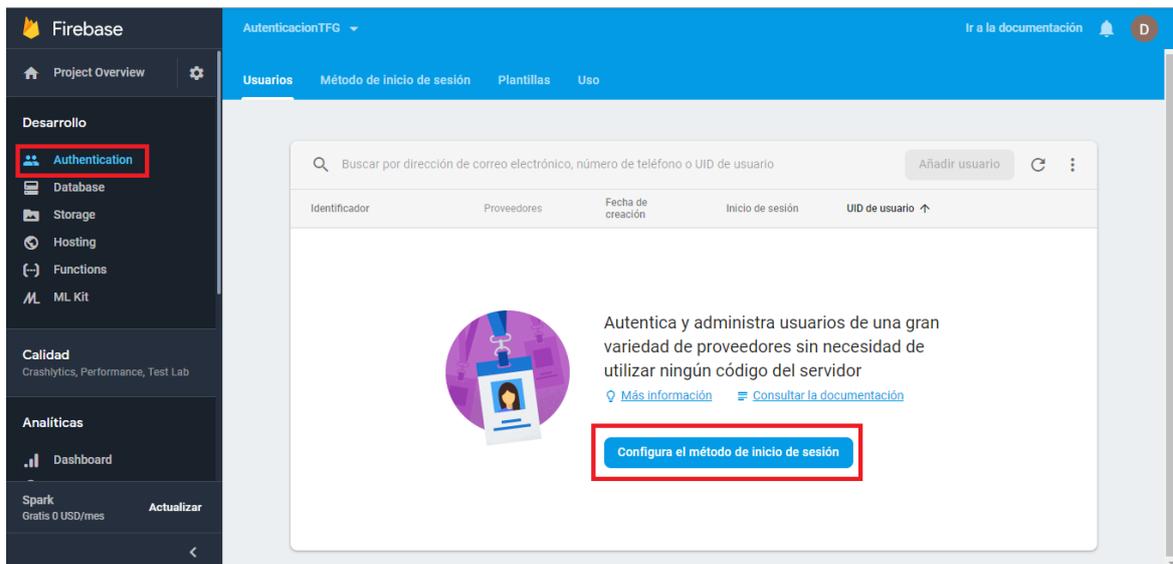


Figura 3.25. Pantalla principal del servicio de Autenticación de Firebase

A continuación, aparece un listado con todas las posibilidades que se le ofrecen al programador para configurar el sistema de inicio de sesión de su aplicación. Dicho listado, se puede observar en la Figura 3.26. Para el caso de la aplicación desarrollada únicamente se ha habilitado la opción de Correo electrónico/contraseña.

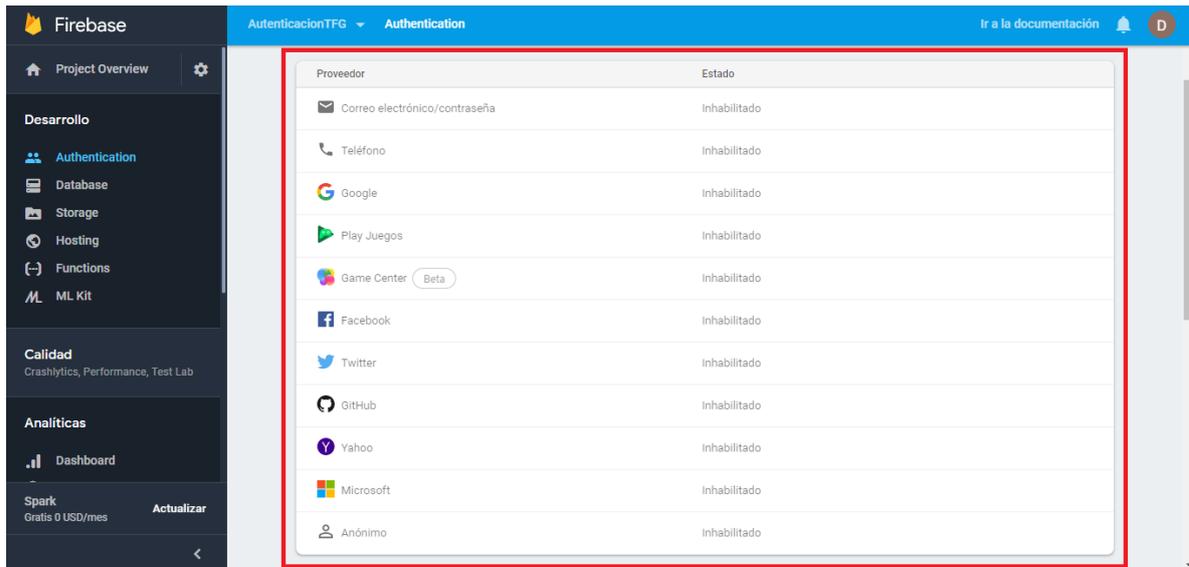


Figura 3.26. Listado con todas las posibilidades de configuración del sistema de autenticación.

Una vez configurado el método de inicio de sesión de los usuarios de la aplicación, desde Android Studio se deben seguir los pasos que detalla el asistente de Firebase. A continuación, se exponen los pasos que se han seguido en el desarrollo del proyecto, los cuales son los mismos que los que ofrece el asistente de Android Studio pero con alguna modificación:

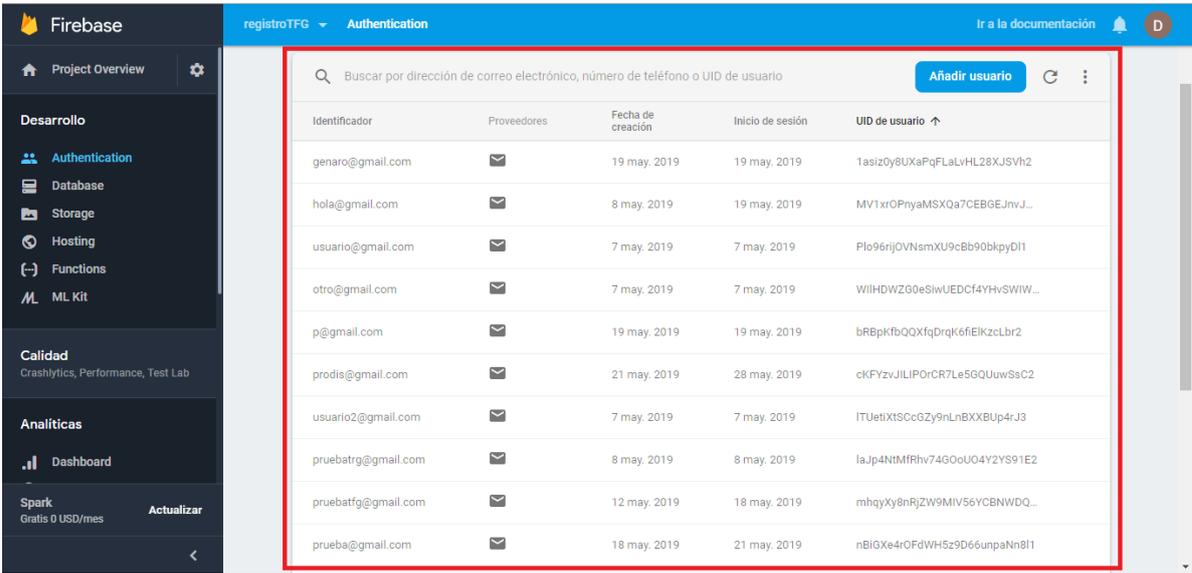
- 1) Declarar la instancia de un objeto `FirebaseAuth` y obtener su instancia en el método `onCreate()`.
- 2) Declarar un escuchador de cuando son pulsados los botones de ‘Registrar usuario’ y de ‘Login’.
- 3) En el escuchador de ‘Registrar usuario’ se realiza una llamada a un método el cual en primer lugar comprueba si los campos de correo y contraseña están vacíos mediante `isEmpty()`. En caso de que alguno de ellos esté vacío, el sistema avisará al usuario de que no se ha podido realizar el registro.

Posteriormente, una vez que la persona que maneja el dispositivo ha rellenado de forma correcta el campo de correo electrónico y de contraseña, mediante el método `createUserWithEmailAndPassword()` de la clase `FirebaseAuth`, se inicia el proceso del registro del nuevo usuario. A dicho método se le añade un escuchador `addOnCompleteListener()` para cuando haya finalizado, comprobar si se ha realizado de forma correcta el registro o si por el contrario ha habido un fallo, por ejemplo de conexión a internet. También se puede dar el caso de que el usuario ya haya sido registrado en otra ocasión. Entonces, se le avisa al dueño del dispositivo que no se ha podido realizar el registro debido a que el usuario ya existe, en cuyo caso simplemente debería pulsar en el botón ‘Login’.

- 4) En el escuchador del botón ‘Login’ se lleva a cabo un proceso muy parecido al anterior paso. Se hace una llamada a un método el cual en primer lugar comprueba si los campos de correo electrónico y contraseña están vacíos. En el caso de que alguno de ellos o los dos lo estén, el sistema realizará un aviso mediante un `Toast` a la persona que maneja el dispositivo.

Si ambos campos han sido completados, es entonces cuando se hace una llamada al método *signInWithEmailAndPassword()* de la clase *FirebaseAuth* para iniciar el proceso de login. A dicho método se le añade un escuchador *addOnCompleteListener()* para cuando haya finalizado, comprobar si se ha realizado de forma correcta el login o si por el contrario ha surgido un fallo, por ejemplo de conexión a internet o que los campos de correo electrónico y contraseña no se corresponden con ningún usuario registrado en el sistema. En el caso de que correo electrónico y contraseña sean correctos, el usuario puede comenzar a hacer uso de los juegos de la aplicación.

Finalmente, en la Figura 3.27. se muestra a modo de ejemplo la consola Firebase del servicio de autenticación generada a partir del registro de algunos usuarios.



The screenshot shows the Firebase Authentication console for a project named 'registroTFG'. The interface is in Spanish. On the left, there is a navigation menu with options like 'Project Overview', 'Desarrollo' (Development), 'Calidad' (Quality), and 'Analíticas' (Analytics). The main area displays a table of registered users. The table has columns for 'Identificador' (Identifier), 'Proveedores' (Providers), 'Fecha de creación' (Creation date), 'Inicio de sesión' (Last sign-in), and 'UID de usuario' (User ID). A search bar at the top allows filtering by email, phone number, or UID. A blue button labeled 'Añadir usuario' (Add user) is visible in the top right of the table area.

Identificador	Proveedores	Fecha de creación	Inicio de sesión	UID de usuario ↑
genaro@gmail.com	✉	19 may. 2019	19 may. 2019	1asiz0y8UXaPqFLaLVL28XJSVh2
hola@gmail.com	✉	8 may. 2019	19 may. 2019	MV1xrOPnyaMSXQa7CEBGEJnvJ...
usuario@gmail.com	✉	7 may. 2019	7 may. 2019	Pl096rjOVNsmXU9cBb90bkpyDl1
otro@gmail.com	✉	7 may. 2019	7 may. 2019	WlHdWZG0eSiwUEDCf4YHvSWIW...
p@gmail.com	✉	19 may. 2019	19 may. 2019	bRBpKfbQQXfqDrqK6fElkzcLbr2
prodis@gmail.com	✉	21 may. 2019	28 may. 2019	cKFYzYJILIPOrCR7Le5GQuwSSc2
usuario2@gmail.com	✉	7 may. 2019	7 may. 2019	lTUetiXtSc6Zy9nLnBXXBUp4rJ3
pruebatrg@gmail.com	✉	8 may. 2019	8 may. 2019	laJp4NTMFRhv74G0oU04Y2YS91E2
pruebatrg@gmail.com	✉	12 may. 2019	18 may. 2019	mhyXy8nrRjZW9MIV56YCBNWDQ...
prueba@gmail.com	✉	18 may. 2019	21 may. 2019	nBiGXe4rOfdWH5z9D66unpaNn8l1

Figura 3.27. Usuarios registrados en la aplicación desarrollada

3.7 Almacenamiento de información

La aplicación desarrollada cuenta con tres partes de almacenamiento de información muy importantes. La primera se trata del lugar donde se almacenan todas las preguntas, respuestas y referencias a las imágenes correspondientes a cada juego. La segunda está compuesta por la forma de guardar el nombre del usuario que ha iniciado sesión para posteriormente, en la tercera parte, almacenar la puntuación conseguida en cada juego y las opiniones que presenta sobre la aplicación ese mismo usuario.

3.7.1 Base de datos SQLite

Tal y como se detalla en la sección 2.3. del presente documento, los motores de bases de datos SQLite son capaces de gestionar bases de datos de gran tamaño de forma estable y eficiente. Es por ello por lo que son utilizadas en la mayoría de las aplicaciones móviles.

En este caso, para la aplicación Android desarrollada también se ha utilizado este motor para almacenar todas las cuestiones, respuestas posibles, respuestas correctas y referencias a las imágenes de cada cuestión, además de las respuestas que puede dar el sistema en un cuadro de dialogo con el usuario que maneja el dispositivo móvil.

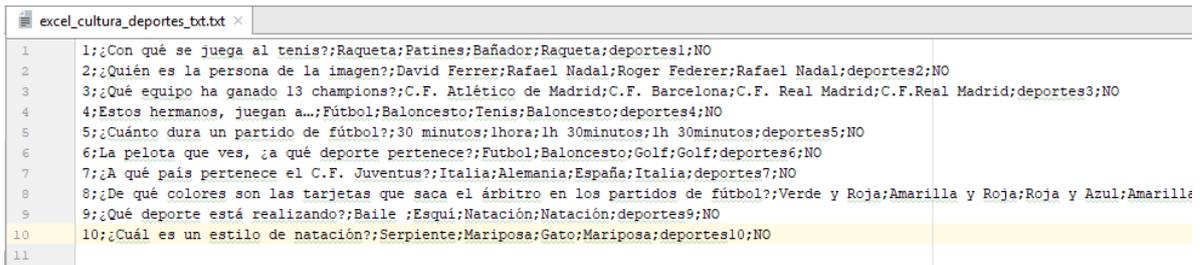
En las secciones 2.3.2 y 2.3.3 se definen dos métodos distintos para inicializar los valores en las tablas de la base de datos. En uno de ellos se leen los datos de un archivo *.txt* almacenado en el subdirectorio *raw* creado a partir de una tabla en Excel y en el otro la información a almacenar se define directamente en el editor de código de Android Studio y se inserta en la tabla que corresponda. Pues bien, para el desarrollo de la aplicación se ha hecho uso de los dos métodos.

- **Lectura de información desde el editor de código de Android Studio (sección 2.3.2):** en la aplicación hay un momento en el que el sistema pregunta al usuario que le ha parecido el último juego al que ha jugado o qué le parece la aplicación en sí. Mediante este método de lectura de información se han inicializado las posibles respuestas que puede dar el sistema a la opinión del usuario. En la Figura 3.28 se muestra una captura de pantalla del código correspondiente.

```
valores.put(COL_TEXTO, "¡Hola! ¿Qué te parece la aplicación?"); //put(primaryKey, tipo de dato que corresponda con la columna)
valores.put(COL_RESPUESTA_PP, "¡Me alegro mucho!");
valores.put(COL_RESPUESTA_P, "¡Genial!");
valores.put(COL_RESPUESTA_NEU, "");
valores.put(COL_RESPUESTA_N, "Intentaré mejorarla");
valores.put(COL_RESPUESTA_NN, "Vaya... Siento que no te guste");
```

Figura 3.28. Inicialización de los valores de la tabla correspondiente de la base datos SQLite a través del editor de código

- **Lectura de información por medio de un archivo *.txt* (sección 2.3.3):** mediante este método de lectura de información se ha procedido a almacenar en las tablas del gestor de datos las preguntas de cada uno de los juegos de la aplicación, las posibles respuestas, la respuesta correcta de cada cuestión y en algunos casos, la referencia a la imagen correspondiente (almacenada en *Drawable*) de la respectiva pregunta. Se observa en la Figura 3.29 un archivo *txt* desde donde se obtiene la información.



Line	Content
1	1;¿Con qué se juega al tenis?;Raqueta;Patines;Bañador;Raqueta;deportes1;NO
2	2;¿Quién es la persona de la imagen?;David Ferrer;Rafael Nadal;Roger Federer;Rafael Nadal;deportes2;NO
3	3;¿Qué equipo ha ganado 13 champions?;C.F. Atlético de Madrid;C.F. Barcelona;C.F. Real Madrid;C.F.Real Madrid;deportes3;NO
4	4;Estos hermanos, juegan a...;Fútbol;Baloncesto;Tenis;Baloncesto;deportes4;NO
5	5;¿Cuánto dura un partido de fútbol?;30 minutos;lhora;lh 30minutos;lh 30minutos;deportes5;NO
6	6;La pelota que ves, ¿a qué deporte pertenece?;Fútbol;Baloncesto;Golf;Golf;deportes6;NO
7	7;¿A qué país pertenece el C.F. Juventus?;Italia;Alemania;España;Italia;deportes7;NO
8	8;De qué colores son las tarjetas que saca el árbitro en los partidos de fútbol?;Verde y Roja;Amarilla y Roja;Roja y Azul;Amarilla
9	9;¿Qué deporte está realizando?;Baile ;Esquí;Natación;Natación;deportes9;NO
10	10;¿Cuál es un estilo de natación?;Serpiente;Mariposa;Gato;Mariposa;deportes10;NO
11	

Figura 3.29. Fichero.txt que inicializa los valores de la tabla correspondiente de la base de datos SQLite

3.7.2 SharedPreferences

La clase SharedPreferences contiene un conjunto de métodos que dan la posibilidad de guardar y recuperar de forma sencilla un par de clave-valor. Las dos únicas restricciones que tiene este método de almacenado es, por un lado, que el valor a conservar debe ser de tipo primitivo, es decir, *int*, *String*, *boolean*, *float*, etc. Por otro lado, la información almacenada solamente se conserva hasta que la aplicación es desinstalada o cerrada[6].

A pesar de que se podría haber utilizado el método que ofrece el asistente de Firebase para recuperar el nombre del usuario que ha iniciado sesión, se ha utilizado la clase SharedPreferences de tal manera que se recupera el texto que se escribe en el campo de correo electrónico para almacenarlo como variable *String*. En la Figura 3.30. se puede observar el procedimiento que se recomienda en el asistente de autenticación de Firebase.

6 Access user information

If a user has signed in successfully you can get their account data at any point with the `getCurrentUser` method.

```
FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
if (user != null) {
    // Name, email address, and profile photo Url
    String name = user.getDisplayName();
    String email = user.getEmail();
    Uri photoUrl = user.getPhotoUrl();

    // Check if user's email is verified
    boolean emailVerified = user.isEmailVerified();

    // The user's ID, unique to the Firebase project
    // authenticate with your backend server, if you
    // FirebaseUser.getToken() instead.
    String uid = user.getUid();
}
```

Figura 3.30. Procedimiento que ofrece Firebase para obtener el nombre del actual usuario

En la sección 3.6. se detallan los pasos seguidos para introducir en la aplicación un sistema de autenticación de usuarios. En el paso número cuatro, se hace referencia a un método que se encuentra en el interior del escuchador que

avisa en el momento en el que se completa el proceso de Login. En dicho método, si se ha logrado acceder a la cuenta del usuario existente de forma satisfactoria, se lleva a cabo el guardado del nombre de usuario de la siguiente manera:

```
SharedPreferences pref_usuarios;  
int pos=email.indexOf("@");  
user=email.substring(0,pos);  
pref_usuarios = getSharedPreferences("prefs_usuarios", 0);  
SharedPreferences.Editor editor = pref_usuarios.edit();  
editor.putString("nombre_usuario",user);  
editor.commit();
```

Mediante el método *indexOf()* se obtiene la posición del carácter '@' introducido en el campo de correo electrónico con el objetivo de separar el nombre del usuario del dominio. Por ejemplo, si la persona que maneja el dispositivo móvil introduce 'pruebaTFG@gmail.com', en el *int pos* se almacena la posición del carácter '@', y en el *String user* se guarda por medio de *substring()* todos los caracteres que hay desde la posición 0 de *email* hasta la posición donde se encuentra '@'. Seguidamente, se recupera una instancia de la clase *SharedPreferences* y de la clase *Editor*. A través del método *putString()* se introduce la pareja de clave-valor correspondiente al nombre del usuario. Finalmente, se llama al método *commit()* para que se efectúe de forma satisfactoria el guardado.

Una vez que ya ha sido conservado el nombre del usuario que está utilizando la aplicación, el proceso de recuperación de la información es muy sencillo. Nuevamente se ha de crear una instancia de la clase *SharedPreferences* para posteriormente con el método *getString()*, en este caso, y la pareja de clave-valor, obtener la información.

3.7.3 Firebase. Realtime Database o base de datos en tiempo real:

En la aplicación Android desarrollada en este proyecto se ha incluido una tercera variante mediante la cual se va a almacenar información. Se trata del servicio de bases de datos en tiempo real que ofrece Firebase. En la base de datos que se cree, se procederá a guardar tanto las puntuaciones que consiga como las opiniones de cada juego que tengan los usuarios registrados de manera independiente. Es decir, el usuario A tendrá unas puntuaciones y opiniones que se almacenan de forma paralela a las puntuaciones y opiniones del usuario B.

Tal y como se especifica en la sección 2.5.2 del documento, tras haber conectado la aplicación a una cuenta Firebase, haber introducido las dependencias correspondientes y haber sincronizado el proyecto para que las dependencias tengan las versiones correctas, se ha de configurar el proyecto Firebase desde la web [39].

Una vez seleccionado el proyecto con el que se ha conectado desde Android Studio y desde el que se está trabajando, se accede a la pestaña de base de datos



‘Database’. En la Figura 3.30 se muestra la pantalla que se debería de visualizar después de acceder a ‘Database’, donde se puede ver que Firebase expone información sobre otro sistema de conservación de información en tiempo real, Cloud Firestore. Sin embargo, en el caso de la aplicación desarrollada se ha hecho uso del servicio Realtime Database.

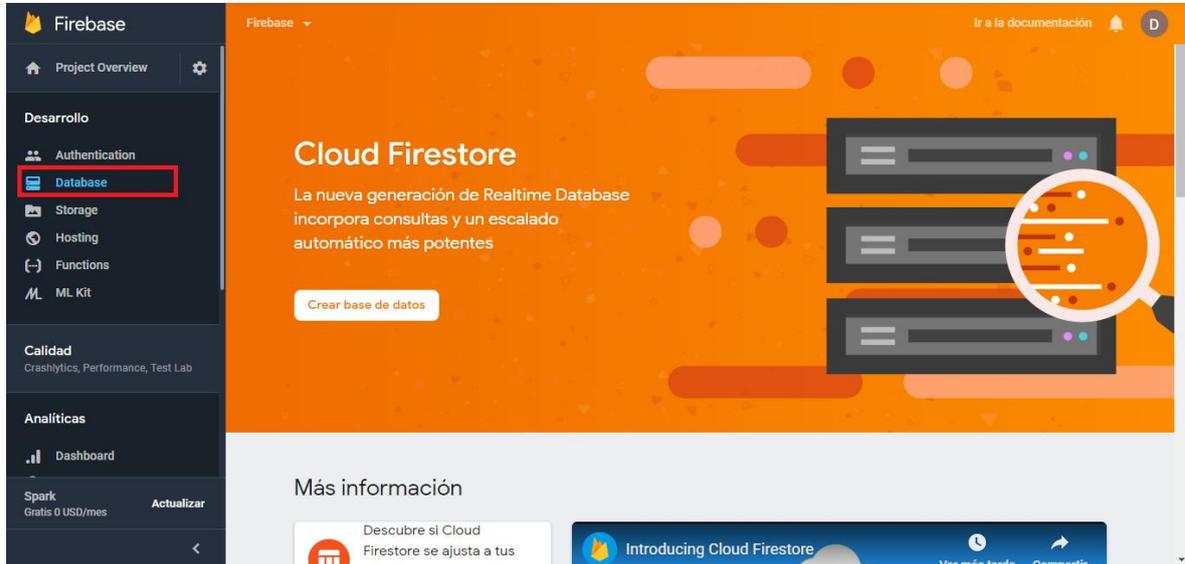


Figura 3.31. Pantalla tras acceder a la sección ‘Database’

Desplazando la barra hacia abajo, se puede encontrar el apartado referido a Realtime Database desde donde se inicia la configuración de la base de datos, como se observa en la Figura 3.32.

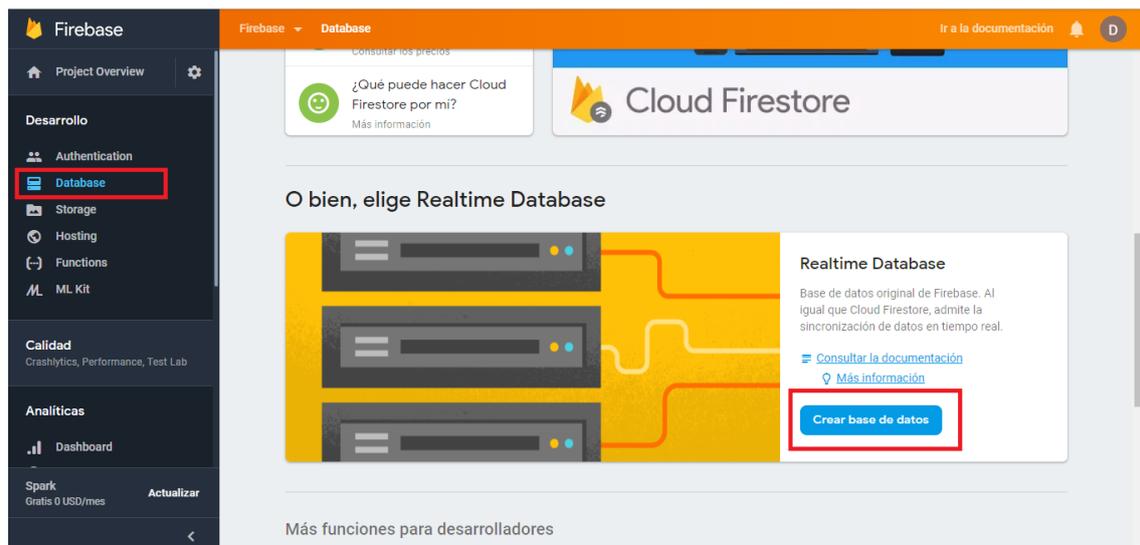


Figura 3.32. Apartado de Realtime Database en la sección ‘Database’

Al hacer clic en el botón señalado en la Figura 3.32 de ‘Crear base de datos’, aparece una ventana donde se debe configurar las reglas de seguridad de la base

de datos. Es decir, hay que definir si las operaciones de escritura y lectura de terceros están bloqueadas o no. En este caso, se ha configurado para que las operaciones de escritura y de lectura estén permitidas para cualquier usuario que tenga la referencia de la base de datos. En la Figura 3.33 se puede ver dicha ventana desde donde se configuran las opciones de seguridad.

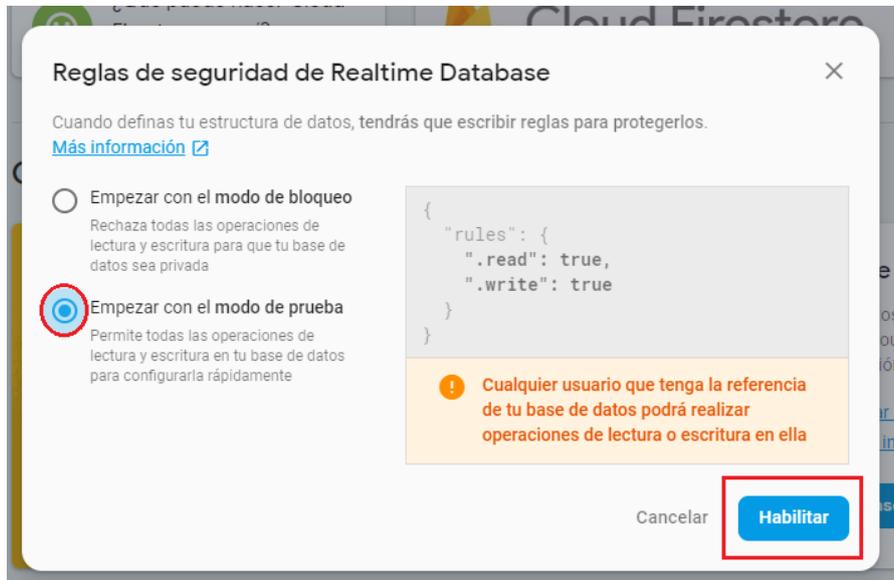


Figura 3.33. Ventana de configuración de las reglas de seguridad de la base de datos en tiempo real de Firebase

Una vez habilitadas las opciones de seguridad en la base de datos, aparece la pantalla que se muestra en la Figura 3.34, donde se puede observar que su estructura se encuentra vacía.

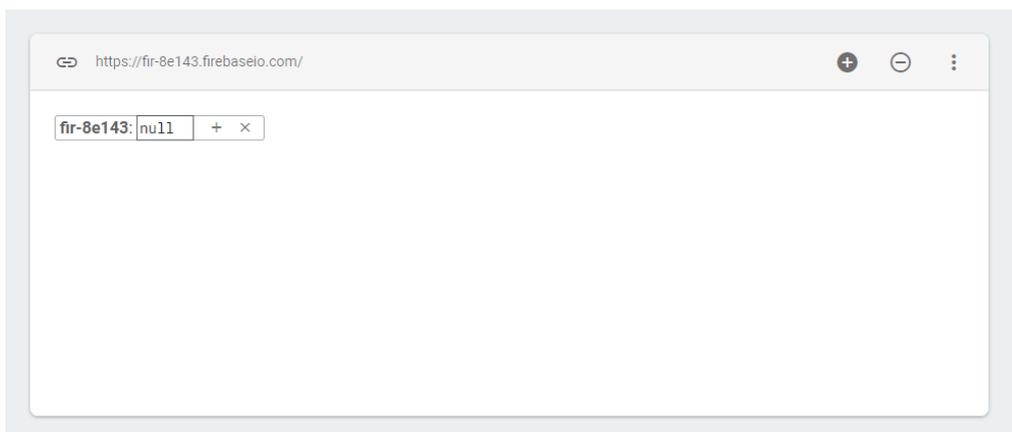


Figura 3.34. Base de datos de Firebase vacía

Estando la base de datos ya preparada para insertar información, desde Android Studio se abre el asistente de Firebase de la sección 'Realtime Database'. En el desarrollo de la aplicación móvil se han seguido los pasos que se detallan

en dicho asistente, pero con algunas modificaciones con el fin de adaptarlo. El procedimiento que se ha llevado a cabo ha sido:

- 1) Se ha creado una clase 'Usuario' la cual contiene todos los parámetros que define a un usuario de la aplicación: las puntuaciones y las opiniones de los juegos.
- 2) En la actividad donde se registran o realizan el login los usuarios, cuando se lleva a cabo la acción de login de forma satisfactoria, se obtiene una instancia de la base de datos de Firebase. A dicha referencia, se le añade el escuchador *addListenerForSingleValueEvent()* tal y como se detalla en el paso número cinco del asistente de Firebase de Android Studio.

En el interior del escuchador hay que sobre escribir el método *onDataChange(DataSnapshot dataSnapshot)* donde *dataSnapshot* es el nodo referencia de la base de datos y a través de un bucle *for* se chequea de entre sus nodos hijos si el usuario que está accediendo a la aplicación es un usuario que se acaba de registrar o ya tenía cuenta. En el caso de que se acabe de registrar en el sistema de autenticación descrito en la sección 3.6., se debe hacer un registro del usuario en la base de datos en tiempo real de dicho usuario con todos los parámetros que se definen en el paso 1. Por ejemplo, el Usuario 4 se acaba de registrar en el sistema de autenticación de Firebase y, por tanto, no existe en la estructura de Realtime Database. A continuación, se debe realizar su registro de forma paralela en dicha base datos con todos los parámetros de puntuaciones y opiniones de los juegos que le corresponden. En la Figura 3.34 se puede observar la estructura que tendría una base de datos en tiempo real de Firebase en la que hay registrados tres usuarios con sus respectivos parámetros.

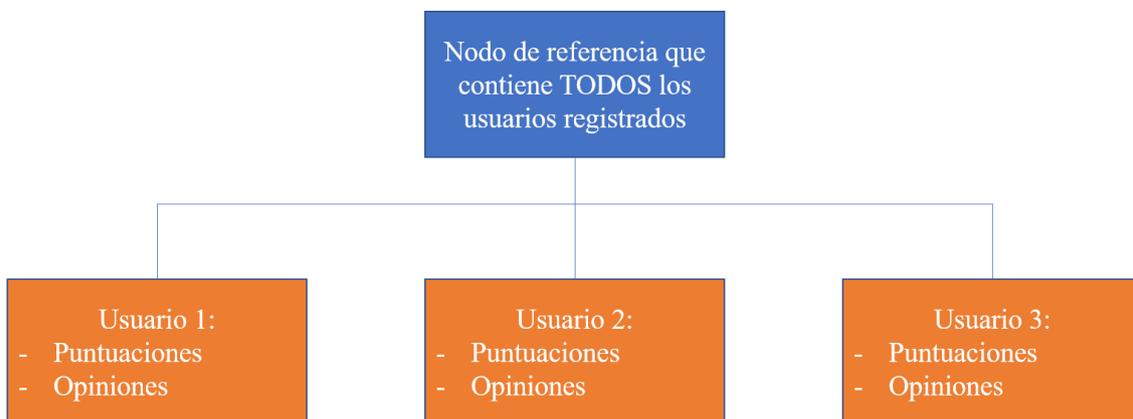


Figura 3.35. Estructura de base de datos de Firebase

- 3) Como se ha comentado en el paso 2, en el caso de que al buscar el usuario que realiza el login no se encuentre entre los hijos del nodo de referencia, se debe llevar a cabo el registro del usuario. Para ello se define un usuario de la clase Usuario (paso 1) con todos sus parámetros de puntuaciones inicializados a cero y todos los parámetros de opiniones inicializados a "". Al nodo de referencia hay que agregarle un hijo con el nombre del usuario, obteniéndolo como se especifica en la sección 3.7.2. del documento a través de *SharedPreferences*. Mediante el método *child()* pasando como

parámetro el nombre del usuario, se crea el nodo hijo y mediante `setValue()` pasando como parámetro el objeto de la clase Usuario se le inicializan los atributos. Este último paso se puede observar en la siguiente línea de código donde `Usuario` es el nodo de referencia, `user` es el nombre del usuario que realiza el login y `alumno_nuevo` es el objeto de la clase Usuario con todos sus atributos inicializados.

```
Usuario.child(user).setValue(alumno_nuevo);
```

A continuación, en la Figura 3.36 a modo de resumen se muestra de forma esquemática los procesos que lleva a cabo el sistema en la pantalla de autenticación.

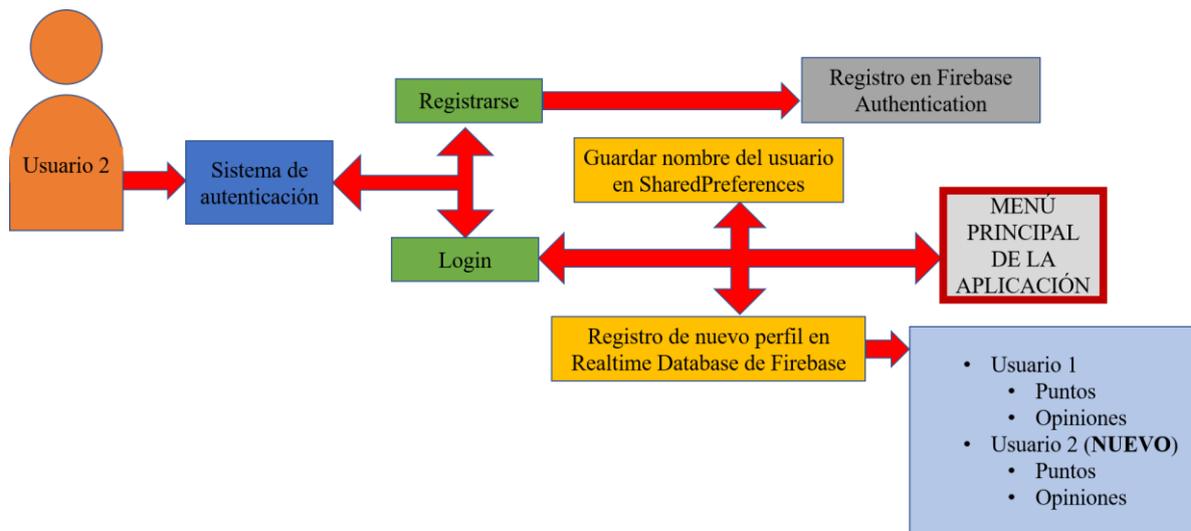


Figura 3.36. Esquema de los procesos llevados a cabo en la pantalla de registro de la aplicación

En algunos de los menús de la aplicación desarrollada, existe una barra de progreso que marca al usuario la cantidad de veces que ha conseguido superar un juego. En algunos de los juegos, cada diez aciertos o puntos, la barra de progreso avanza un décimo del total. Es por eso por lo que, igual que se insertan datos en la base de datos en tiempo real de Firebase, es necesario obtener en determinadas partes del código los puntos que ha conseguido el usuario en un determinado juego.

Cuando el jugador desea abandonar un juego, presiona el botón *back* de su teléfono móvil, apareciendo una ventana donde se pregunta si se está seguro de abandonar la actividad. En el momento en el que se pulsa en el botón correspondiente al abandono de la actividad, es cuando se debe obtener la información almacenada en la base de datos para que en las barras de progreso del menú se vean reflejados los puntos obtenidos. Para ello, se utiliza un método muy parecido al descrito anteriormente:

- 1) En el escuchador de presión del botón de abandono de actividad se añade un escuchador `addListenerForSingleValueEvent()` a una referencia de la base de datos de Firebase.

2) En el interior del escuchador hay que sobre escribir el método `onDataChange(DataSnapshot dataSnapshot)` donde `dataSnapshot` es el representante del nodo referencia de la base de datos. A través de un condicional `if` se chequea si de entre todos los nodos hijos que hay, el nodo con el nombre del usuario que ha estado jugando es uno de ellos. Por tanto, se ha de obtener el nombre del usuario que entró en la aplicación, es decir, tal y como se menciona en la sección 3.7.2. se ha de recuperar de la clase `SharedPreferences` el nombre del usuario que fue almacenado al realizar el login. Seguidamente, tras comprobar que es un nodo hijo, a través de un bucle `for` se busca de entre todos ellos el nodo con el mismo key o nombre de usuario. A continuación, en a), se muestra la línea de código correspondiente al condicional `if` en el que se comprueba la existencia del nodo hijo y en b) las líneas de código correspondiente al bucle `for` donde se busca el nodo hijo que tiene el mismo nombre o key que el nombre del usuario que ha estado jugando.

```
a) if (dataSnapshot.hasChild(pref_usuarios.getString("nombre_usuario", "")))

b) for (DataSnapshotchildSnapshot: dataSnapshot.getChildren()) {
    key_usuario=childSnapshot.getKey();

    if (key_usuario.equals(pref_usuarios.getString("nombre_usuario", ""))) {
        datos_puntos=childSnapshot.getValue(Usuario.class);
        break;
    }
}
```

- 3) Una vez localizado el nodo, se obtienen todos los datos que contiene mediante el método `getValue()`, tal y como se muestra en b) del paso 2.
- 4) Finalmente, para que desde la actividad donde se muestran las barras de progreso se puedan recuperar las puntuaciones del jugador, se ha vuelto a almacenar la información o puntos de cada juego a través de la clase `SharedPreferences`. De esta manera, es accesible desde cualquier lugar de la aplicación.

Cuando el usuario accede a la aplicación, el procedimiento seguido para obtener las puntuaciones de su perfil y poder representar de forma correcta todas las barras de progreso o contadores de puntos es muy similar al descrito. Se realiza la localización del nodo del usuario, se obtienen sus puntuaciones y se almacenan a través de `SharedPreferences` para que sean accesibles desde todas las actividades de la aplicación.

En la Figura 3.37 se muestra a modo de resumen un esquema donde se observan los procedimientos seguidos para obtener los puntos de la base de datos en tiempo real de Firebase.



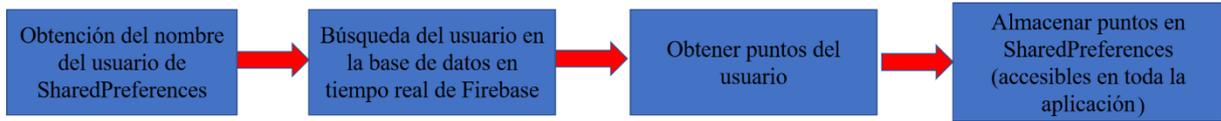


Figura 3.37. Proceso para obtener puntos de la base de datos en tiempo real de Firebase

Para finalizar la sección de bases de datos en tiempo real del documento, solamente queda detallar cuál sería el proceso en caso de que se quisiese actualizar algún valor de la base de datos de Firebase:

- 1) En primer lugar, se debe obtener una instancia de la base de datos.
- 2) Seguidamente, a través del método *child()* se debe obtener como referencia el nodo de cuyo usuario se va a actualizar los valores, es decir, el nodo con el nombre del usuario activo. Nuevamente para ello se hace uso de la clase *SharedPreferences* para obtener el nombre del usuario que se ha almacenado en el proceso de login. A continuación, se muestra la línea de código que obtiene dicha referencia.

```
final DatabaseReference hopperRef =
Usuario.child(pref_usuarios.getString("nombre_usuario", ""));
```

- 3) Por otro lado, se debe obtener una referencia del nodo de información que se va a actualizar, es decir, los puntos del JuegoX del UsuarioY. De igual forma que en el paso 2, mediante el método *child()* se obtiene dicha referencia. Sin embargo, a esta segunda referencia se le debe añadir un escuchador *addListenerForSingleValueEvent()* para sobre escribir el método *onDataChange(DatasSnapshot snapshot)*. En la siguiente línea de código se muestra este paso.

```
Usuario.child(pref_usuarios.getString("nombre_usuario",
"")).child("puntos_fracciones3").addListenerForSingleValueEvent(new
 ValueEventListener() {...})
```

- 4) Se define un objeto de la clase *HashMap* que se le asigna una pareja de clave-valor mediante el método *put()*. Por último, con el método *updateChildren()* y pasando como parámetro el objeto de tipo *HashMap*, se lleva a cabo la actualización de un atributo de un usuario concreto.

```
final Map<String, Object> hopperUpdates = new HashMap<>();
hopperUpdates.put("puntos_fracciones3", snapshot.getValue(Integer.class)+1);
hopperRef.updateChildren(hopperUpdates);
```

Se muestra en la Figura 3.38 en forma de diagrama los procesos seguidos para llevar a cabo la actualización de una puntuación.

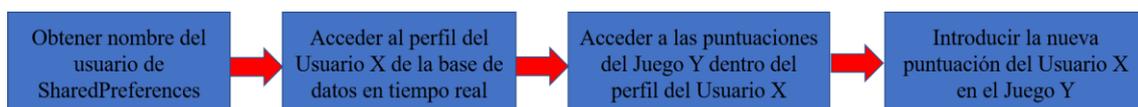


Figura 3.38. Proceso seguido para actualizar una puntuación en la base de datos en tiempo real de Firebase

En la Figura 3.39 se pueden observar cada uno de los nodos de los que se ha hablado anteriormente. Por un lado, en verde queda recuadrado el nodo de referencia general ('Usuarios'). Por otro lado, en el siguiente nivel se pueden observar todos los usuarios registrados ('Drg', 'a' y 'b'). En rojo queda recuadrado

el usuario con nombre 'b', el cual contiene sus respectivos puntos y opiniones, que quedan recuadrados en azul.

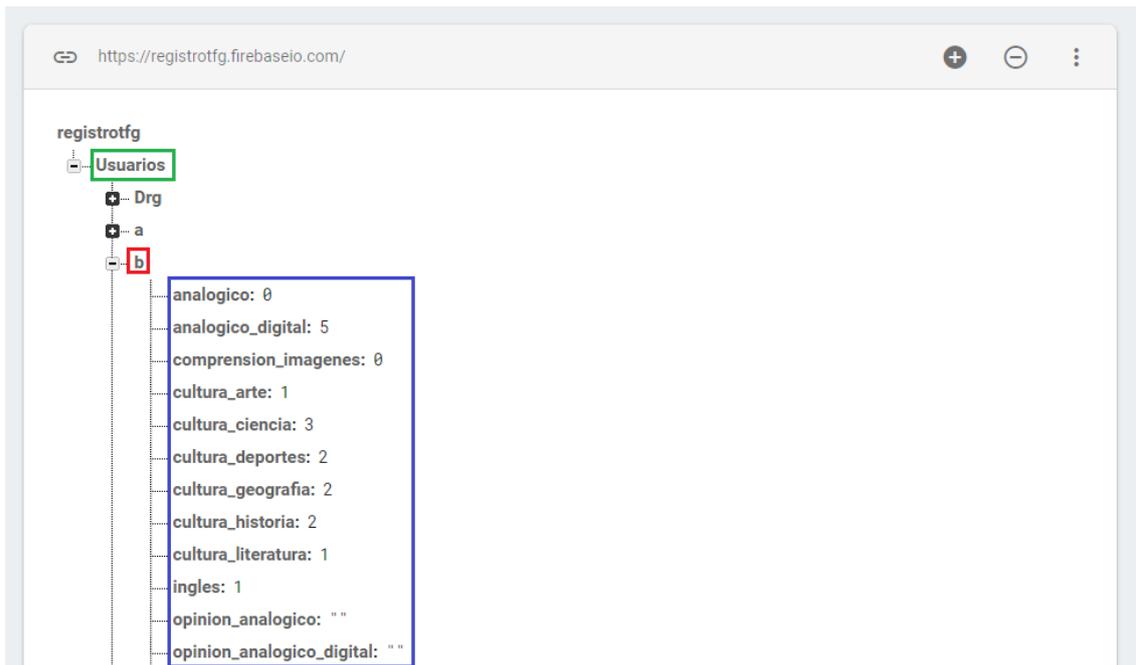


Figura 3.39. Realtime Database con datos actualizados

3.8 MeaningCloud

Se ha pretendido idear una forma original de conocer la opinión de los usuarios tanto de los juegos implementados como de la aplicación en general. Consiste en una pequeña conversación con el personaje o mascota de la aplicación.

Cada cierto tiempo, cuando el usuario se encuentre en el menú principal, se abrirá un cuadro de diálogo en el que dicho personaje o bien le pregunte al usuario por el último juego al que jugó o bien por su opinión acerca de la aplicación. Además, para que el desarrollador de la aplicación conozca el criterio del usuario, todas las respuestas quedarán almacenadas en la base de datos en tiempo real de Firebase mediante los métodos descritos en la sección 3.7.3. del documento.

Por otro lado, con el objetivo de que, tras la respuesta escrita por el usuario, el personaje o mascota de la aplicación reaccione con una nueva contestación, se ha utilizado la herramienta de análisis de sentimiento ofrecida por MeaningCloud. Cuando el usuario pulse el botón enviar, el texto que haya escrito en el campo de respuesta al personaje será almacenado para llevar a cabo una consulta a los servidores de MeaningCloud y conocer su polaridad. Dicha polaridad tiene la posibilidad de ser: muy positiva (P+), positiva (P), neutral (NEU), negativa (N), muy negativa (N+) o sin sentimiento (NONE).

En la base de datos SQLite detallada en la sección 3.7.1 se han almacenado las posibles respuestas que puede dar el personaje en función de la polaridad de la respuesta del usuario a la cuestión. De esta manera, tras la consulta al servidor de

MeaningCloud, la mascota de la aplicación realizará una contestación para terminar la conversación basándose en la polaridad obtenida.

A continuación, se detallan los pasos que se han seguido para realizar la consulta al servicio de análisis de sentimientos de MeaningCloud:

- 1) Darse de alta en MeaningCloud desde su página web [67] y obtener una clave para poder realizar peticiones al servidor.
- 2) Se ha importado el ejemplo 'Network Connect' que ofrece Android Studio desde *File>New>Import Sample* y se ha adaptado su clase NetworkFragment, la cual es la encargada de llevar a cabo la conexión con el servicio y obtener el resultado, y la interfaz DownloadCallback para realizar la conexión con MeaningCloud.
- 3) Se ha almacenado a través de la clase SharedPreferences el texto con el que el usuario ha respondido al personaje en la conversación.
- 4) Desde la clase principal de la aplicación desarrollada se ha realizado una llamada al método de la clase NetworkFragment encargado de hacer la petición a MeaningCloud recuperando con SharedPreferences el texto a analizar y utilizando la clave obtenida en el paso 1.

3.9 Esquema resumen del sistema

Para finalizar el presente capítulo de la memoria, se muestra en forma de gráfico en la Figura 3.40 la relación que existe entre todos los servicios utilizados.

En primer lugar, a partir del sistema de autenticación de usuarios se obtiene el nombre del usuario que se dispone a utilizar la aplicación.

Además, como se ha detallado, la aplicación está compuesta de tres partes importantes: los juegos, las barras de progreso que muestran al usuario lo entrenado que está en un área y un cuadro de diálogo con el personaje de la aplicación.

Los juegos obtienen las preguntas, las respuestas posibles, la respuesta correcta y, en ocasiones, la referencia a una determinada imagen de la base de datos SQLite. Cuando el usuario escoge o escribe una respuesta para la pregunta, se vuelve a consultar en la base de datos SQLite para saber cuál es la respuesta correcta. Si la persona que maneja el móvil ha acertado, a su vez se debe actualizar la base de datos en tiempo real de Firebase.

Las barras de progreso siempre deben estar actualizadas. Es por ello por lo que, si en algún momento se lleva a cabo un cambio en las puntuaciones de un usuario en la base de datos en tiempo real de Firebase, automáticamente se debe actualizar las barras de progreso del usuario.

Por último, el sistema de conversación con el personaje o mascota de la aplicación está incluido con el objetivo de saber la opinión del usuario. Cuando dicho personaje pregunte al usuario por su opinión sobre algún juego o sobre la aplicación en sí, dicha respuesta se almacena en la base de datos en tiempo real y de esta manera el desarrollador es conocedor de su criterio. Por otro lado, para que la conversación no concluya con una simple respuesta del usuario, dicha contestación es enviada al

analizador de sentimiento de MeaningCloud obteniendo como resultado su polaridad. En función de la polaridad, el personaje responderá una frase u otra, obtenidas de la base de datos SQLite, para concluir la conversación.

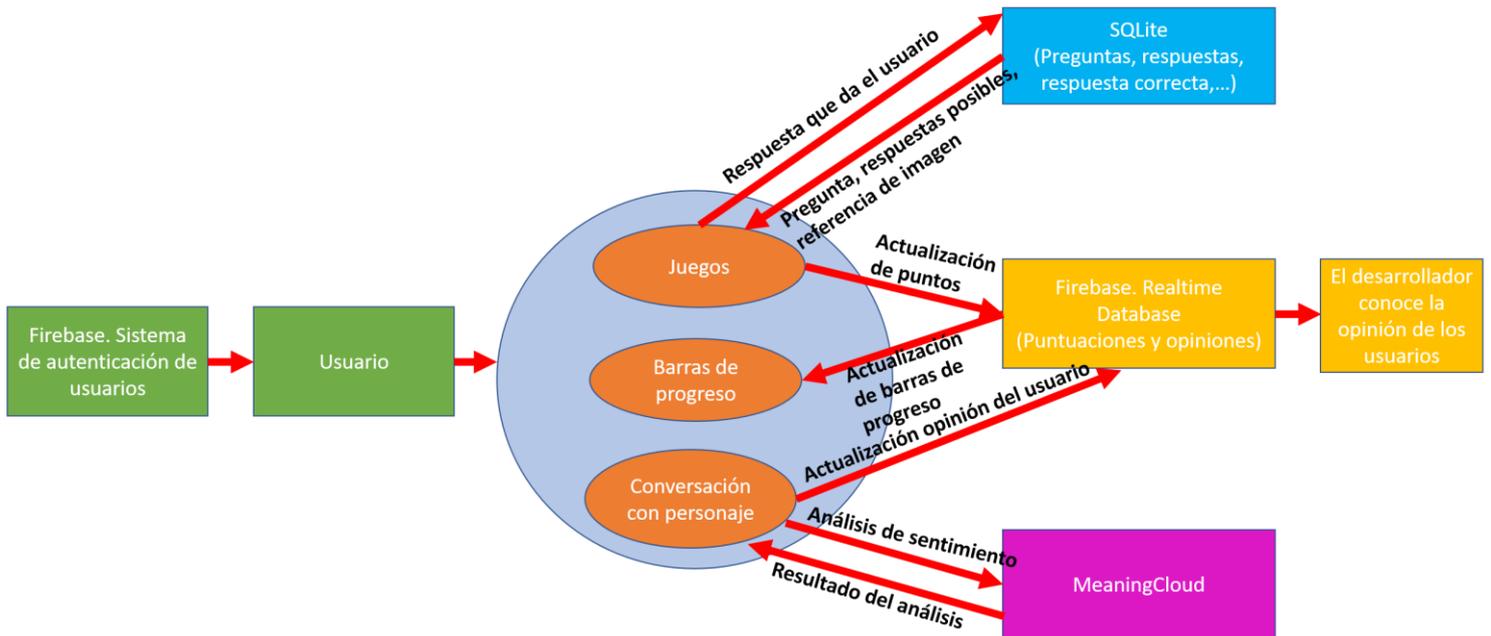


Figura 3.40. Esquema resumen de la aplicación

CAPÍTULO 4. Descripción detallada del sistema

En el capítulo 4 del documento se procede a la descripción detallada de la aplicación desarrollada durante los últimos meses. Además de realizar una explicación del origen del nombre que se le ha asignado a la aplicación, se parte de la pantalla inicial y se accede a cada una de las secciones disponibles definiendo las tareas que lleva a cabo el sistema a la vez que el usuario maneja el dispositivo móvil.

4.1 Origen del nombre del sistema

La diversidad funcional es un término que pretende sustituir las palabras ‘discapacidad’ o ‘minusvalía’ por ejemplo, las cuales en ocasiones son percibidas de forma negativa [82]. La diversidad funcional se refiere a que las personas tenemos cada uno unas capacidades distintas a las de los demás y que deben ser gestionadas de tal manera que no se produzca exclusión o discriminación social [83].

Se ha querido hacer un guiño a dicho concepto y por ello se ha asignado el nombre de ‘DiverApp’ a la aplicación desarrollada.

4.2 Inicio de la aplicación

Al pulsar el icono de la aplicación comienza a cargarse el sistema. A modo de presentación del trabajo desarrollado, en primer lugar, se muestra una pantalla formada por un gif con el nombre de la aplicación, el icono y el nombre del autor. En la Figura 4.1. se observa dicha pantalla inicial.

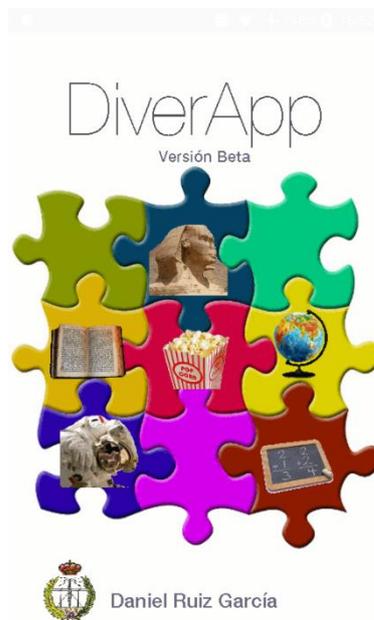


Figura 4.1. Inicio de DiverApp

4.3 Autenticación del usuario

Seguidamente, aparece la pantalla correspondiente al sistema de autenticación de los usuarios donde aparece el personaje de la aplicación. En ella, se debe insertar en los campos de texto que aparecen un correo electrónico y una contraseña. Si la persona que maneja el dispositivo móvil aún no ha sido registrada en el sistema, primero debe pulsar en el botón ‘Registrar usuario’. Si ya ha sido registrado el usuario, debe pulsar el botón ‘Login’ para acceder al menú principal.

Además, por medio de un botón se puede acceder a un aviso de la privacidad que declara el desarrollador de la aplicación al usuario.

Se observan todos los elementos descritos en la Figura 4.2.

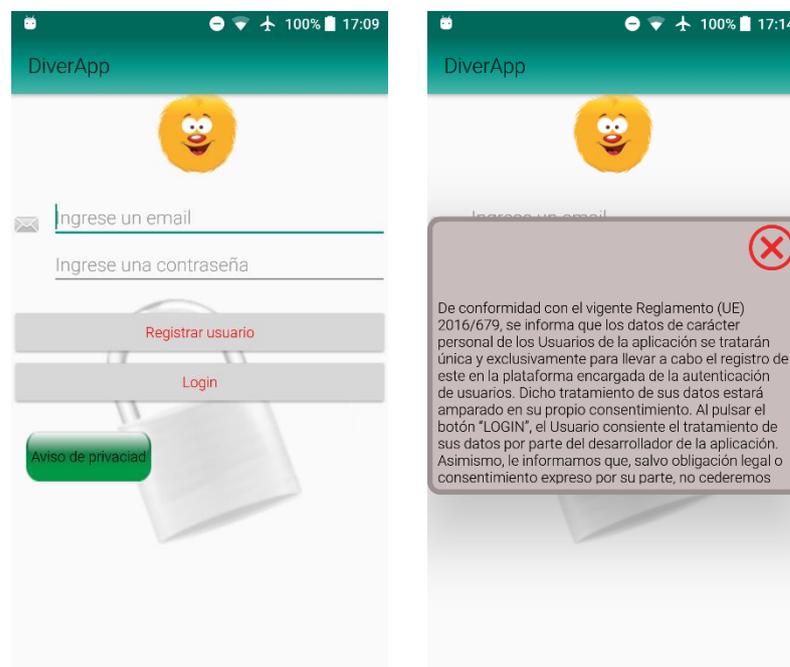


Figura 4.2. Sistema de autenticación de usuarios de la aplicación desarrollada.

En el momento en el que se introduce un correo electrónico y una contraseña y se pulsa en el botón ‘Registrar usuario’, el sistema se encarga de conectar con Firebase Authentication para llevar a cabo el registro del usuario.

Por otro lado, cuando correo electrónico y contraseña han sido ingresados y se pulsa el botón ‘Login’, el sistema comprueba si es la primera vez que el usuario utiliza la aplicación. En caso de ser la primera vez, se accede a la base de datos en tiempo real y se genera un nuevo hijo del nodo raíz ‘Usuarios’ con todos los atributos de puntos y opiniones inicializados a cero y a “” respectivamente. En caso de no ser la primera vez que el usuario accede a la aplicación, quiere decir que su perfil ya ha sido creado en la base de datos en tiempo real de Firebase y, por tanto, se accede a él y se obtienen sus puntos. Además, todos los puntos, haya sido la primera vez que se accede

a la aplicación con un usuario o no, son almacenados en las preferencias de usuario de la clase SharedPreferences para ser utilizados en pantallas posteriores.

Finalmente, el nombre del usuario que accede a la aplicación también es almacenado a través de SharedPreferences ya que en cada uno de los juegos es necesaria la actualización de los puntos de un determinado usuario en la base de datos en tiempo real de Firebase.

4.4 Menú principal

Una vez finalizada la autenticación del usuario, se accede al menú principal donde se observan cinco botones que hacen referencia a cada una de las áreas que se pretenden practicar con esta aplicación: lengua, matemáticas, inglés, cultura y autogestión. Además, a cada botón se le ha asignado un nombre referenciado al término de diversidad funcional, al igual que el de la aplicación; Diver Lengua, Diver Mates, Diver Trivial y Diver English. El quinto botón posee en su interior al personaje de la aplicación.

También cuenta en la esquina superior derecha, en la barra de herramientas (*toolbar*), un botón mediante el cual un usuario puede salir de su cuenta y, en la superior izquierda se ha implementado un menú deslizante, el cual actualmente no está operativo, desde donde está previsto que en futuras versiones de la aplicación el usuario pueda acceder a sus respectivos resultados, tutoriales, configuración, información de la aplicación, sea capaz de compartirla, etc. La disposición de todos los elementos mencionados se muestra en la Figura 4.3.

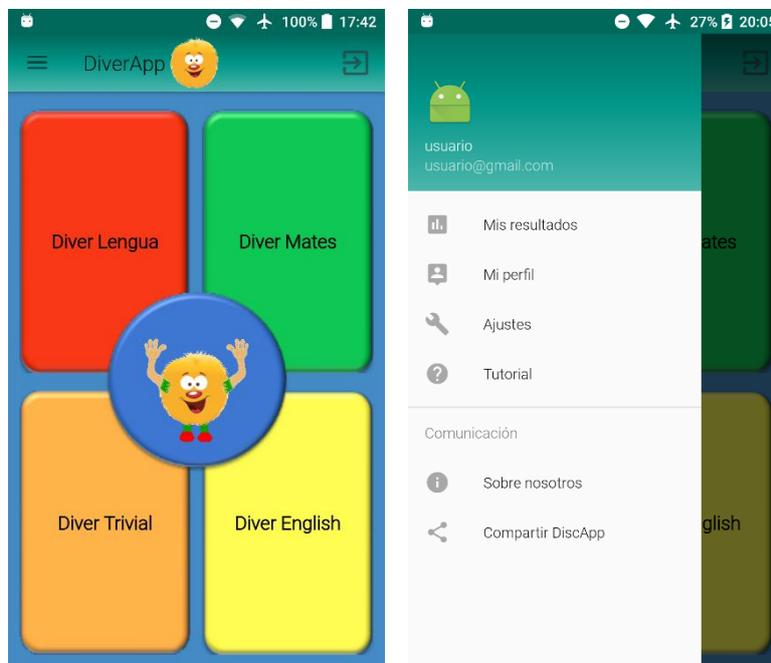


Figura 4.3. Menú principal de la aplicación desarrollada.

En el momento en el que se pulsa cualquiera de los cinco botones se accede a las diferentes áreas de trabajo, las cuales se describen en las secciones siguientes del presente capítulo 4.

Además, en la pantalla del menú principal se ha ideado una manera original de obtener la opinión de los usuarios acerca de la aplicación. Cada sesenta segundos sin pulsar ningún botón de la pantalla del menú principal, aparece un cuadro de dialogo con el personaje de la aplicación donde se pregunta al usuario que le parece el sistema desarrollado. Se puede observar en la Figura 4.4. dicho cuadro de diálogo de la aplicación.



Figura 4.4. Cuadro de diálogo con el personaje de la aplicación desarrollada.

Cuando el usuario escribe para contestar al personaje y pulsa el botón ‘Enviar’, dicho texto es almacenado y, por medio de una conexión con el servicio de análisis de sentimiento de MeaningCloud, se obtiene la polaridad de la respuesta enviada. En función del resultado obtenido, se accede al gestor de datos SQLite y se escoge la respuesta adecuada. En la Figura 4.5 se puede observar un ejemplo distinto de contestación por parte del personaje al que se ofrece en la Figura 4.4.



Figura 4.5. Cuadro de diálogo con el personaje de la aplicación desarrollada.

4.5 Diver Lengua

Al pulsar en el botón que hace referencia a esta sección desde el menú principal de la aplicación, tras una transición entre actividades, se accede a una lista donde aparecen los juegos con los que el usuario puede practicar el área lingüística. Actualmente cuenta con tres juegos diferentes los cuales tienen asignados cada uno de ellos una barra de progreso que se actualiza en función de las veces que se logren los objetivos de la actividad. La Figura 4.6. muestra dicha lista de juegos además del botón 'back' que posee la barra de herramientas (*toolbar*).



Figura 4.6. Lista de juegos de 'Diver Lengua'

Las barras de progreso de cada uno de los juegos se actualizan en función de la información que se ha obtenido de la base de datos en tiempo real de Firebase y que se ha almacenado en la clase SharedPreferences.

Pulsando en cada uno de los ítems de la lista, se accede a los juegos que se detallan en las próximas secciones. Por otro lado, pulsando en el botón atrás, tras una transición se vuelve a la pantalla del menú principal.

4.5.1 Diver Frases

El usuario, arrastrando, debe ordenarla palabras para formar frases. En la Figura 4.7. se muestra dicha actividad.



Figura 4.7. Juego 'Diver Frases'

Se han definido las regiones de 'Resp1', 'Resp2', 'Resp3', 'Resp4', 'Resp5' y 'Resp6' como las posibles áreas donde poder arrastrar las palabras. Cada una de estas regiones y cada palabra tiene un código. Al arrastrar una palabra sobre una región si ambos tienen el mismo código, se marca como correcto. Si no coinciden, el arrastre ha sido fallido.

4.5.2 Diver Textos

En primer lugar, aparece un breve texto el cual el usuario debe leer atentamente ya que tras pulsar el botón 'Continuar' se le realizarán preguntas con dos respuestas posibles y deberá responder de forma correcta el máximo posible durante treinta segundos. En la Figura 4.8. se muestran las dos pantallas correspondientes a esta actividad.

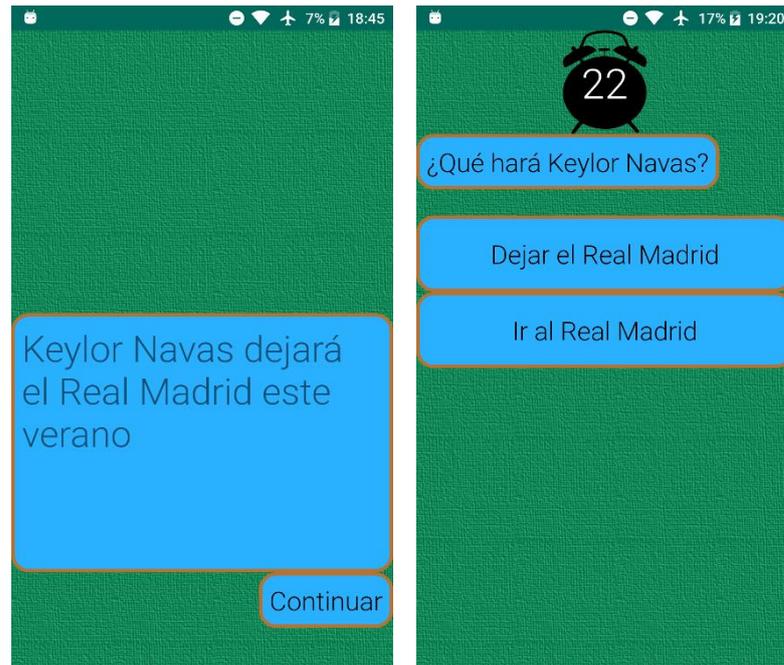


Figura 4.8. Juego ‘Diver Textos’

En el momento en el que se inicia el juego, se accede al gestor de datos SQLite para obtener un texto determinado. Dicho texto es leído por el sistema a través de un sintetizador de texto a voz y al pulsar el botón ‘continuar’ se cargan sus respectivas preguntas, las cuales también son sintetizadas. Aparece un reloj que realiza la cuenta atrás de treinta segundos y el usuario, cuando selecciona una de las dos respuestas posibles, se lleva a cabo una nueva consulta al sistema de almacenamiento SQLite para comprobar si es correcto o no. En el caso de que sea correcto la respuesta cambia su *background* y se pone en color verde. Si es incorrecta, el *background* de la respuesta cambia a color rojo.

Si el usuario consigue responder de forma correcta las cinco preguntas que se llevan a cabo en el tiempo establecido, gana un punto y por tanto hay que actualizar la base de datos en tiempo real de Firebase. Por otro lado, también hay que actualizar los datos guardados en SharedPreferences ya que se toma como referencia lo almacenado a través de dicha clase para actualizar las barras de progreso del usuario.

Por último, al concluir la cuenta atrás del tiempo establecido, aparece un cuadro de dialogo con el número de respuestas correctas y preguntando al jugador si desea jugar de nuevo o prefiere salir al menú principal.

4.5.3 Diver Memoria

Al iniciar el juego ‘Diver Memoria’ el usuario debe estar atento a una secuencia de cinco imágenes que se le muestran ya que posteriormente, en la siguiente pantalla, de entre doce palabras debe seleccionar las que estén relacionadas lo imágenes visualizado. Ambas pantallas se pueden observar en la Figura 4.9.

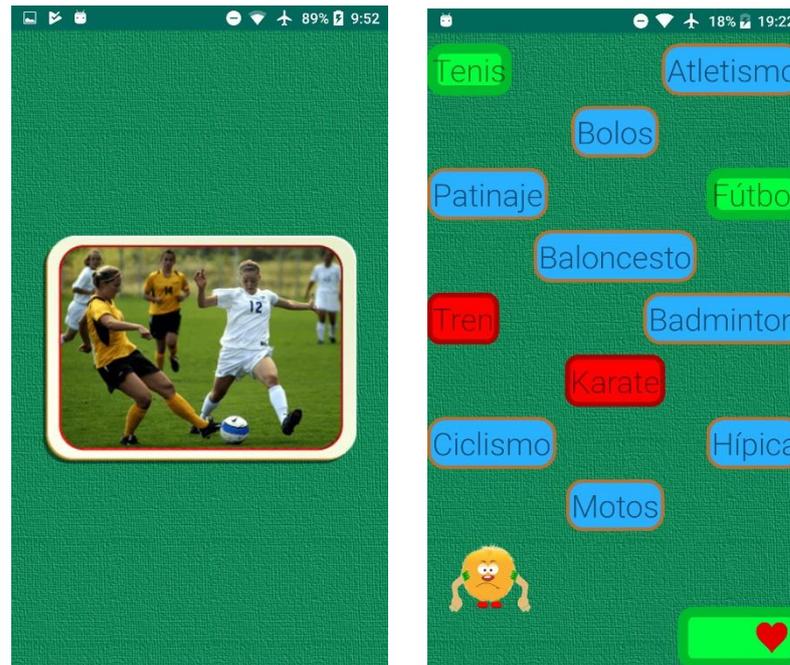


Figura 4.9. Juego ‘Diver Memoria’.

Además, como se puede ver, el usuario cuenta con tres vidas. En el momento que pulsa en una de las palabras, se accede a la base de datos SQLite para comprobar si es una de las cinco respuestas correctas posibles. En el caso de que lo sea, el *background* de la palabra cambia a otro *shape* o forma de color verde. Si por el contrario la palabra seleccionada no se corresponde a ninguna de las palabras correctas, su *background* cambia de *shape* a uno de color rojo y, además, pierde una de las tres vidas que tiene. Cuando se han perdido las tres vidas aparece un cuadro de diálogo donde se le pregunta al jugador si desea volver a jugar o prefiere volver al menú principal. Por otro lado, si se ha logrado seleccionar las cinco palabras correctas sin perder las tres vidas aparece otro cuadro de dialogo realizando la misma pregunta al usuario.

Al igual que en los casos anteriores, si el usuario ha logrado el objetivo del juego, sus puntos deben ser actualizados en la base de datos en tiempo real de Firebase para que si deja de jugar en la aplicación y en otro momento vuelve a realizar el login, se carguen todos sus datos. A su vez, para actualizar las barras de progreso se actualizan dichos puntos en las preferencias del usuario de la clase *SharedPreferences*.

4.6 Diver Mates

Desde el menú principal de la aplicación, el cual ha sido detallado en la sección 4.4., se puede acceder al área de entrenamiento de las matemáticas. Al pulsar en el botón ‘Diver Mates’ aparece una pantalla con una lista de todos los juegos disponibles, al igual que la detallada en la sección 4.5 Dicha lista junto con las barras de progreso de cada juego se puede observar en la Figura 4.10.



Figura 4.10. Lista de juegos de 'Diver Mates'

Cada una de las barras de progreso se actualiza en función de los datos almacenados en las preferencias del usuario de SharedPreferences, los cuales son obtenidos cuando se realiza el login y son actualizados cada vez que el jugador logra un objetivo determinado.

Pulsando sobre los *ítems* de la lista se accede a cada uno de los juegos.

4.6.1 Diver operaciones

Aparece una operación matemática sencilla (suma o resta) en la parte superior de la pantalla. Por otro lado, las respuestas posibles aparecen y desaparecen en posiciones aleatorias de la pantalla y el jugador debe estar atento para pulsar la que crea que es correcta. En la Figura 4.11 se pueden ver los elementos que componen la actividad.

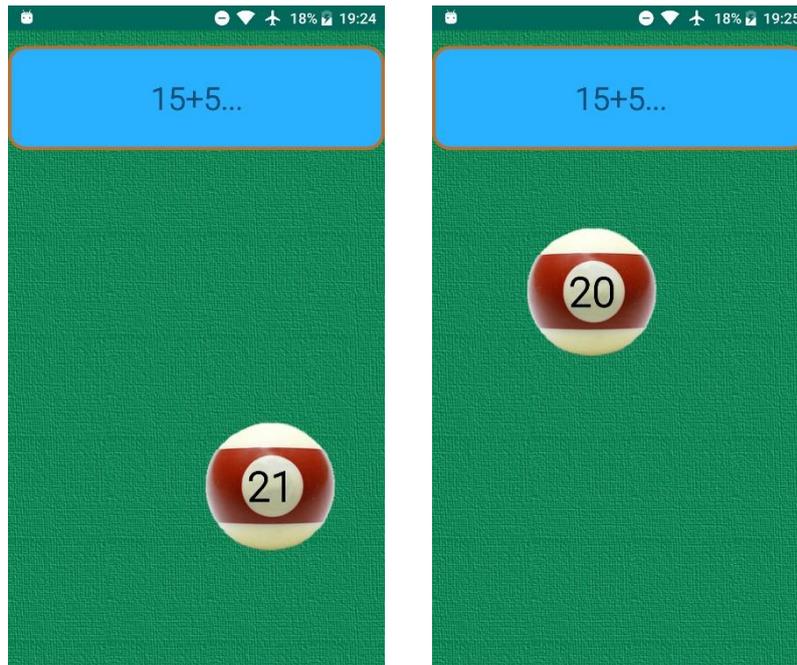


Figura 4.11. Juego ‘Diver Mates’.

Cada una de las cinco posibles respuestas que aparecen y desaparecen han sido cargadas de la base de datos SQLite. Cuando el usuario selecciona una respuesta, se vuelve a acceder a la base de datos SQLite y se comprueba si es correcto. Sea correcta o incorrecta la selección, tras comprobar si lo es, aparece una nueva operación con sus nuevas posibles respuestas.

Por otro lado, cada diez operaciones respondidas correctamente, se produce una actualización en los puntos del usuario. Es decir, como se ha dicho anteriormente, se actualizan los datos de las preferencias de usuario de SharedPreferences ya que es de donde se obtienen los puntos para restablecer las barras de progreso. Además, hay que recordar que también hay que actualizar los puntos en la base de datos en tiempo real de Firebase para que cuando el usuario cierre la aplicación y la quiera volver a abrir pueda continuar con sus respectivos puntos.

4.6.2 Diver Fracciones 1

Se trata de preguntas tipo test con tres respuestas posibles sobre las secciones que se ven representadas en una imagen. En la Figura 4.12 se observa dicho juego.



Figura 4.12. Juego ‘Diver Fracciones 1’.

Seleccionando la respuesta que se cree que es correcta, se produce un efecto de *fade in/fade out* durante unos pocos segundos y finalmente cambia su *background* por un color verde en caso de que sea correcto y rojo en caso de que sea incorrecto.

Para obtener la pregunta, las respuestas posibles y comprobar si la selección es correcta se accede a la base de datos SQLite. Además, cada cuestión de la tabla de la base de datos SQLite tiene asignado un nombre de imagen. Dicho nombre o referencia es buscado entre todas las imágenes guardadas en el subdirectorio *drawables* y se inserta en la pantalla.

Por último, nuevamente, cada diez aciertos el atributo de puntos de esta actividad del usuario debe ser actualizado, se le debe sumar uno y que se vea reflejado en la barra de progreso. Para ello, se actualizan los puntos almacenados en las preferencias del usuario de SharedPreferences ya que es de donde se obtienen los datos para actualizar las barras de progreso. Por otro lado, al igual que anteriormente se debe modificar la puntuación en la base de datos en tiempo real de Firebase para la próxima vez que el jugador realice login.

4.6.3 Diver Fracciones 2

La dinámica del juego es la misma que el de la sección 4.6.1. del documento. Se realiza una cuestión referente al área de las fracciones y de forma aleatoria aparecen y desaparecen por la pantalla las posibles respuestas que puede seleccionar el jugador. Se puede observar en la Figura 4.13 la actividad que se detalla.



Figura 4.13. Juego ‘Diver Fracciones 2’.

Cada una de las respuestas que aparecen y desaparecen en posiciones de la pantalla aleatorias han sido obtenidas de la base de datos SQLite, al igual que la cuestión que se realiza y la respuesta correcta. Cuando el usuario pulsa en una respuesta, se comprueba si coincide con la respuesta correcta. Haya sido o no haya sido correcta la selección del jugador, posteriormente aparece una nueva cuestión con nuevas respuestas posibles.

Tras haber respondido de forma correcta diez preguntas, se debe actualizar la puntuación del usuario y con ello la barra de progreso del juego. Para ello se le suma uno a la puntuación de la actividad que se almacena en las preferencias del usuario de SharedPreferences. También se ha de actualizar la puntuación del usuario en la base de datos en tiempo real para que cuando en un futuro vuelva a acceder a la aplicación sea posible cargar sus datos en las preferencias de usuario de SharedPreferences.

4.6.4 Diver Fracciones 3

El jugador debe escribir el numerador y el denominador que forman la fracción que representa la sección que se muestra en la imagen, tal y como se muestra en la Figura 4.14.



Figura 4.14. Juego ‘Diver Fracciones 3’

Tras escribir el numerador y denominador correspondiente, se debe pulsar el botón ‘Confirmar’, el cual cambiará su *background* a verde si ha sido correcto y a rojo si es incorrecto.

La cuestión y la respuesta correcta han sido obtenidas de la tabla correspondiente a este juego de la base de datos SQLite. También a partir del nombre o referencia de la imagen de la cuestión obtenida de ella, se busca de entre todas las imágenes almacenadas en el subdirectorio *drawables* la imagen que corresponda.

Por último, al igual que en casos anteriores, cada diez respuestas correctas se ha de actualizar tanto las puntuaciones en las preferencias del usuario de SharedPreferences como las puntuaciones de la base de datos en tiempo real.

4.6.5 Diver Euro 1

Al usuario se le informa de lo que puede llegar a costar comprar un determinado objeto o llevara cabo una determinada acción. A partir de ello, el jugador debe arrastrar monedas y billetes a una caja registradora de tal forma pagué la cantidad exacta del objeto acción en cuestión. En la Figura 4.15 se muestra la disposición de la actividad.



Figura 4.15. Juego 'Diver Euro 1'.

En la zona inferior de la pantalla se ha definido un ViewPager, el cual cuenta con tres páginas distintas. Arrastrando en el sentido que indica la flecha se accede a las distintas páginas que componen el ViewPager. En una de ellas se encuentran todas las monedas y en las dos restantes han sido colocados los billetes.

Se han definido cada una de las imágenes de monedas y billetes como regiones disponibles para llevar a cabo un arrastrado (*drag*). Por otro lado, la imagen de la caja registradora ha sido establecida como la región donde se deben arrastrar las monedas y billetes. Es decir, si el usuario arrastra una moneda y la suelta (*drop*) sobre la caja registradora, el arrastre se habrá efectuado de forma correcta.

A cada uno de los billetes y monedas se le ha asignado un valor, ambos en las mismas unidades. De esta manera, cada vez que se lleva a cabo de forma correcta un arrastre de una moneda o de un billete (*drag and drop*), se suma a un contador de dinero el valor correspondiente. Cuando dicho contador alcance la cantidad exacta por la que se pregunta, se realiza un aviso al usuario de que lo ha logrado y además se debe proceder a la actualización de puntuaciones tanto en las preferencias del usuario de SharedPreferences como en la base de datos en tiempo real de Firebase. En caso de superar la cantidad establecida como precio, también se le notifica al jugador.

4.6.6 Diver Euro 2

Aparecen dos imágenes, en ocasiones cada una de ellas con una cantidad de dinero y en otras ocasiones la misma cantidad de dinero, pero expresada con



monedas o billetes distintos. El usuario debe decidir si ambas cantidades son las mismas o no y para ello debe pulsar le botón ‘SI’o el botón ‘NO’. En la Figura 4.16. se muestra la actividad de la que se habla.

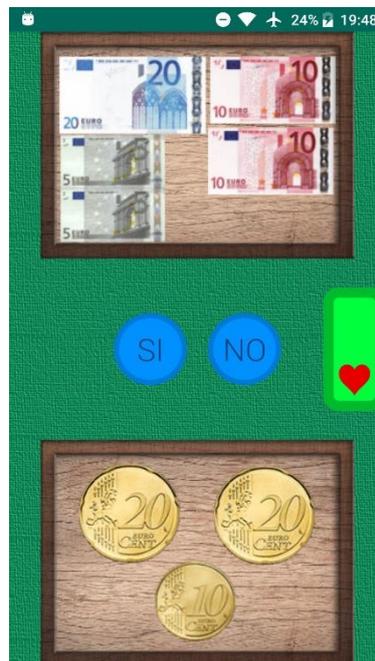


Figura 4.16. Juego ‘Diver Euro 2’

De la tabla correspondiente a este juego de la base de datos SQLite se obtiene tanto las referencias de las imágenes para buscarlas en el subdirectorio e insertarlas *drawables* como la respuesta correcta. Tras pulsar en uno de los dos botones se comprueba si se ha respondido de forma correcta. Si la respuesta ha sido incorrecta, el jugador pierde una de las tres vidas que tiene. Después de haber respondido de forma correcta diez preguntas, se le suma uno a la puntuación del juego almacenada en las preferencias del usuario de SharedPreferences y se actualiza la puntuación en el perfil del usuario de la base de datos en tiempo real de Firebase, para que si en un futuro el usuario vuelve a entrar en la aplicación y realiza el login se carguen de forma correcta sus puntuaciones.

4.7 Diver Trivial

Pulsando el botón ‘Diver Trivial’ desde el menú principal de la aplicación se accede al juego desde el cual se van a reforzar algunos aspectos culturales. Se puede observar una ruleta la cual los usuarios tienen que hacer girar pulsando sobre ella para posteriormente realizar una pregunta tipo test sobre el tema seleccionado. Esta actividad sigue la dinámica del tradicional juego del Trivial. En la parte inferior de la pantalla se lleva la cuenta de las preguntas de cada tema que han sido acertadas y cada tres preguntas acertadas el usuario consigue un punto. La disposición de todos los elementos del juego se pueden observar en la Figura 4.17.





Figura 4.17. Juego 'Diver Trivial'.

Se ha generado una circunferencia la cual ha sido dividida en seis partes. Desde los 30° hasta los 90° se ha asignado la temática de arte, desde lo 90° hasta los 150° los deportes, desde los 150° hasta los 210° la temática de historia, desde los 210° hasta los 270° la de geografía, desde los 270° hasta los 330° la de ciencia y, finalmente, los últimos 60° comprendidos entre los 330° y los 30° se les ha asignado el área de literatura. Por otro lado, el selector de tema ha sido colocado en la posición de 0° .

La circunferencia se hace girar al pulsar sobre ella y cuando se detiene, se realiza una pregunta del área cuya sección de la circunferencia se encuentre en la posición de 0° .

Por otro lado, en la parte inferior de la pantalla se lleva la cuenta de la cantidad de preguntas que han sido respondidas correctamente. Este contador, realiza una función parecida a la de las barras de progreso definidas en las secciones 4.5 y 4.6 Cada vez que se acierta una pregunta de un tema determinado, se suma uno en las puntuaciones almacenadas en las preferencias de usuario de SharedPreferences. Además, también se actualiza la puntuación del juego en el perfil del usuario en la base de datos en tiempo real de Firebase.

4.8 Diver English

Se trata del clásico juego de memoria de destapar cartas buscando parejas. En este caso, la mitad de las cartas al destaparlas muestran una imagen y la otra mitad muestran una palabra en inglés que se relaciona con una imagen. En la Figura 4.18. se observa el juego que se detalla.



Figura 4.18. Juego 'Diver English'.

Al pulsar sobre las cartas que se encuentran boca abajo, se destapan por medio de una animación y muestran una imagen o una palabra en inglés. Cada carta que posee una imagen tiene un código el cual es un número del uno al cinco y cada carta que contiene una palabra en inglés también es poseedora de un código el cual es un número del uno al cinco. El usuario debe destapar una carta y el sistema se detiene a la espera de que se destape una segunda carta. Si la segunda carta descubierta tiene el mismo código que la primera, se habrá encontrado una pareja. Si son distintos los códigos, las cartas se vuelven a dar la vuelta por medio de una animación.

Cada vez que se completa un ejercicio de esta sección se actualizan los puntos del usuario. En este caso no hay ninguna barra de progreso y por lo tanto únicamente se actualiza la puntuación referente al juego del perfil del usuario en la base de datos en tiempo real de Firebase.

4.9 Autogestión

Pulsando en el botón central del menú principal de la aplicación se accede al menú desde donde se trabajan diversos aspectos de la autogestión, como se puede observar en la Figura 4.19.

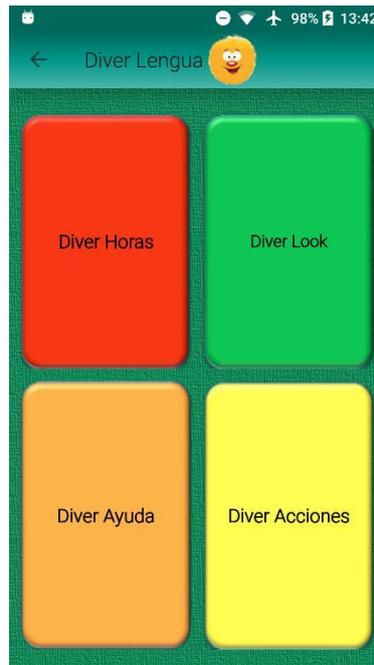


Figura 4.19. Menú de actividades de autogestión.

4.9.1 Diver Horas

En Diver Horas el usuario podrá practicar aspectos relacionados con los relojes y las horas. Al pulsar sobre su botón, se da paso a un menú donde aparecen todas las actividades disponibles junto con su barra de progreso. Actualmente solamente hay dos, las cuales se muestran en la Figura 4.20.

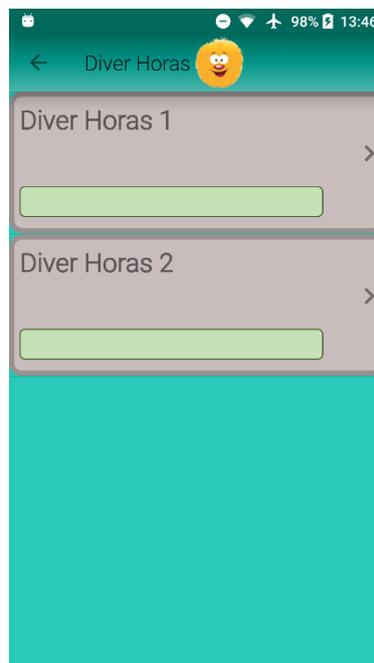


Figura 4.20. Menú de 'Diver Horas'

Nuevamente, cada una de las barras de progreso se actualiza en función de los datos que haya almacenados en las preferencias del usuario de SharedPreferences y por tanto cada vez que el jugador consiga un punto, deben actualizarse.

- **Diver Horas 1**

En la pantalla por un lado aparecen horas representadas en relojes digitales y por otro lado aparecen las mismas horas representadas en relojes analógicos. El usuario debe unir aquellos relojes que marquen las mismas horas. Se muestra un ejemplo de la actividad en la Figura 4.21.

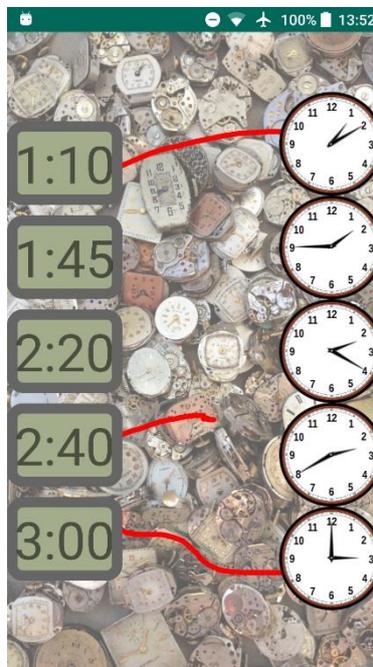


Figura 4.21. Juego 'Diver Horas 1'.

De la tabla correspondiente a este juego de la base de datos SQLite se han obtenido cada una de las cifras de los relojes digitales y un nombre o referencia de los relojes analógicos, de tal manera que dicha referencia es buscada en el subdirectorio *drawables* para insertar la imagen oportuna. Por otro lado, cada reloj, ya sea analógico o digital, tiene asignado un código el cual también es obtenido de la tabla de SQLite.

Para poder realizar los trazados de las líneas se ha utilizado Canvas. Se ha definido cada reloj digital como región a partir de la cual se puede comenzar una línea y cada imagen de reloj analógico como región donde debe acabar dicho trazado. En caso de que no se termine el trazado de la línea en un reloj analógico, automáticamente la línea desaparece. Una vez que se ha completado un trazado, se comprueba si el código del reloj digital donde se comenzó el trazado es el mismo que el código del reloj analógico donde se ha terminado. Si es igual, quiere

decir que es correcto. En caso de que no lo sea, automáticamente la línea o trazado se borra.

Cuando todos los relojes estén relacionados, el usuario habrá concluido el juego y se actualizan las puntuaciones tanto en las preferencias del usuario de SharedPreferences para que las barras de progreso estén correctamente representadas como las puntuaciones del usuario en la base de datos en tiempo real de Firebase, para que en un futuro cuando el usuario vuelva a acceder a la aplicación se puedan cargar de forma correcta sus puntuaciones en las barras de progreso.

- **Diver Horas 2**

En esta sección de la aplicación se realizarán preguntas tipo test al usuario acerca de la hora que está representada en un reloj analógico, tal y como se puede ver en la Figura 4.22.



Figura 4.22. Juego 'Diver Horas 2'.

Se ha obtenido de la tabla correspondiente a este juego de la base de datos SQLite una referencia de la imagen que aparece. Dicha referencia ha sido buscada en el subdirectorio *drawables* y se ha insertado la imagen correspondiente. Por otro lado, tanto la cuestión, como las posibles respuestas y la respuesta correcta también se han obtenido de SQLite.

Cuando el usuario pulsa sobre una de las respuestas, tras un efecto de *fade in/fade out* se marca si la selección ha sido correcta o no. En caso de ser correcta, el *background* de la respuesta se cambia a verde y si es incorrecta, a rojo.

Cada diez aciertos el usuario suma un punto y por tanto se tiene que ver reflejado en la barra de progreso. Para ello, al igual que en los casos anteriores se

suma uno en la puntuación correspondiente al juego tanto en las preferencias del usuario de SharedPreferences como en la base de datos en tiempo real de Firebase.

4.9.2 Diver Ayuda

En este juego se le plantea al usuario un problema para que posteriormente mediante un procedimiento de *drag and drop* seleccione cual es la mejor solución. La interfaz de este juego se puede observar en la Figura 4.23.

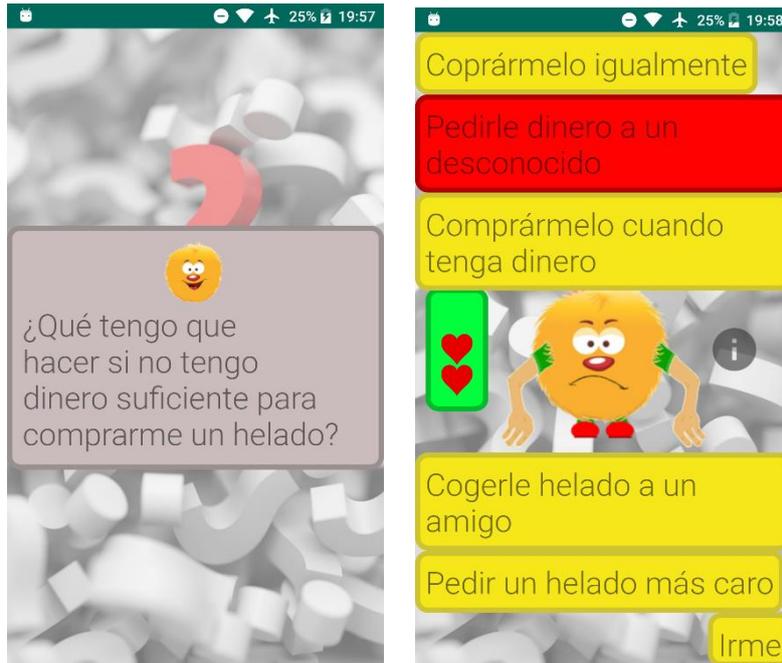


Figura 4.23. Juego 'Diver Ayuda'.

Tanto el enunciado del problema como las respuestas posibles y la respuesta correcta han sido obtenidos de la tabla correspondiente de la base de datos SQLite.

Además, cada respuesta posible ha sido definida como región que está disponible para ser arrastrada y la imagen del personaje de la aplicación como región a la que se pueden arrastrar elementos. De esta manera, el jugador debe desplazar la respuesta que desee al personaje de la aplicación. Si la respuesta es incorrecta, el usuario pierde una vida y la imagen del personaje es la que se muestra en la Figura 4.23. Si la respuesta es correcta, el usuario suma un punto y la imagen del personaje es la que se muestra en la Figura 4.24. Finalmente, cuando el usuario suma un punto, se actualiza la base de datos en tiempo real de Firebase para que en futuros accesos a la aplicación puedan ser cargados los puntos del perfil correspondiente. No hace falta guardar esta puntuación en las preferencias de SharedPreferences ya que no se dispone de barra de progreso o contador de puntos similar en la aplicación.



Figura 4.24. Juego 'Diver Ayuda'.

4.9.3 Diver Acciones

Por último, se cuenta con una actividad en la que los usuarios deben ordenar las acciones para llevar a cabo una tarea determinada. En la Figura 4.25. se muestra la interfaz del juego que se detalla.

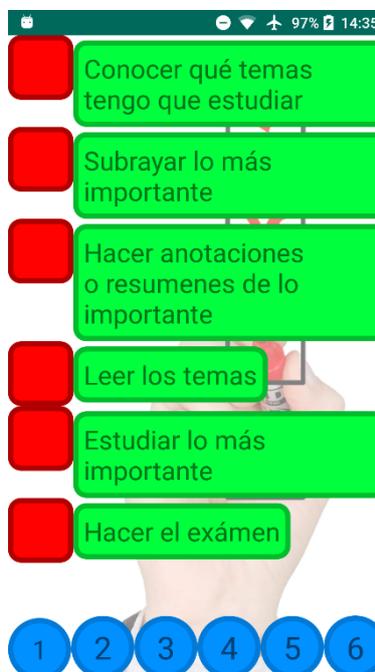


Figura 4.25. Juego 'Diver Acciones'.

De la base de datos SQLite se han obtenido las acciones a ordenar y, además, el código que identifica cada una de ellas. Cada uno de los *shapes* rojos es poseedor de la identificación obtenida y, además, han sido definidos como región a las que es posible arrastrar un elemento. Por otra parte, los

shapes azules han sido definidos como regiones que se pueden arrastrar, al mismo tiempo que tienen otro código asignado. Por tanto, cuando el jugador desplaza los *shapes* azules sobre los rojos, el sistema comprueba si los códigos identificadores coinciden.

Cuando todos los *shapes* azules hayan sido colocados, el juego ha concluido y se actualizan los puntos de la base de datos en tiempo real de Firebase.

4.10 Sistema de puntos

Como se ha explicado, existen algunos menús de la aplicación que contienen unas barras de progreso que se actualizan en función de los aciertos que haya durante un determinado juego. Los sistemas de puntuaciones que se han seguido para actualizar las barras de progreso de los menús han sido:

- Diver Operaciones: cada 10 aciertos consecutivos el usuario suma 1/10 a la barra de progreso del juego.
- Diver Fracciones 1: cada 10 aciertos consecutivos el usuario suma 1/10 a la barra de progreso del juego.
- Diver Fracciones 2: cada 10 aciertos consecutivos el usuario suma 1/10 a la barra de progreso del juego.
- Diver Fracciones 3: cada 10 aciertos consecutivos el usuario suma 1/10 a la barra de progreso del juego.
- Diver Euro 1: cada acierto suma 1/10 a la barra de progresión
- Diver Euro 2: cada 7 aciertos consecutivos el usuario suma 1/10 a la barra de progreso.
- Diver Frases: cada vez que se ordene de forma correcta la frase, se suma 1/10 a la barra de progreso.
- Diver Textos: cada vez que se respondan correctamente las 5 preguntas de un texto, se suma 1/10 a la barra de progreso.
- Diver Memoria: cada vez que se soluciona correctamente el ejercicio, se suma 1/10 a la barra de progreso.
- Diver Horas 1: cada vez que se soluciona correctamente el ejercicio, se suma 1/10 a la barra de progreso.
- Diver Horas 2: cada 10 aciertos consecutivos el usuario suma 1/10 a la barra de progreso del juego.



CAPÍTULO 5. Evaluación de la aplicación

Para llevar a cabo la evaluación de la aplicación desarrollada para el Trabajo Fin de Grado se ha contado con la ayuda de la Fundación Prodis, la cual el día 21 de mayo de 2019 ha dado la posibilidad de ir a la clase de uno de sus programas para personas con discapacidad intelectual a explicar a sus alumnos el proyecto.

La Fundación Prodis tiene como principal objetivo mejorar la calidad de vida de las personas con discapacidad intelectual y sus familias. De forma más concreta, en el programa Avanzas que fue al que se asistió, a sus alumnos con discapacidad intelectual leve se les aplica una formación preuniversitaria con el principal objetivo de lograr un mayor manejo con las nuevas tecnologías y alcanzar una mayor madurez [68].

El programa Avanzas cuenta con dos aulas distintas de diez alumnos aproximadamente cada una y en cada una de ellas se ha impartido una sesión de hora y media. En las sesiones, en primer lugar, se ha realizado una presentación a través de una pizarra digital. En dicha presentación, para que fuese lo más ameno y divertido posible para los chicos y chicas, los alumnos salen a la pizarra a manejar la aplicación a la vez que se realiza una explicación. A continuación, en la Figura 5.1. se muestra la pizarra digital donde se ha realizado la presentación.



Figura 5.1. Aplicación desarrollada en la pizarra digital

En la segunda parte de cada sesión, se presta durante un tiempo los dispositivos móviles con la aplicación instalada desde los que se han estado realizando las pruebas de simulación. Se les ha distribuido en grupos de 3 personas, máximo 4, y se les ha dado total libertad para jugar e investigar todos los apartados de la aplicación. En la Figura 5.2 se puede ver como una alumna del programa Avanzas de Prodis está jugando con la aplicación en un dispositivo móvil prestado.

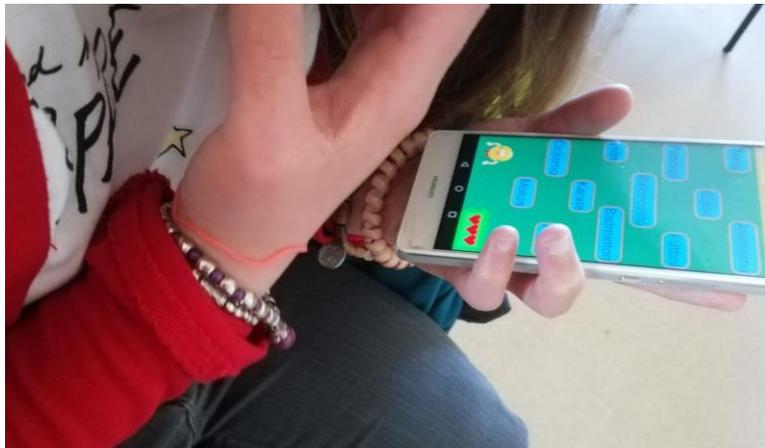


Figura 5.2. Alumno utilizando la aplicación

Por último, para conocer la opinión de los chicos y chicas de la clase y saber qué aspectos se deben mejorar, a través de sus responsables, se les ha facilitado en sus dispositivos móviles personales que utilizan en la fundación una encuesta realizada a través de Formularios Google con preguntas y respuestas muy sencillas. En ellas se pretende evaluar el diseño de la aplicación, la dificultad de los juegos y la autonomía a la hora de utilizarla. En la Figura 5.3 se puede observar el Formulario Google creado.

A screenshot of a Google Form titled "DiverApp". The form is overlaid on a background image featuring a cartoon character with large eyes and a small chalkboard with math problems. The form text includes: "Ayúdanos a saber si la aplicación de móvil que acabas de utilizar te ha gustado o no. Contesta a las siguientes preguntas.", "*Obligatorio", "Dirección de correo electrónico *", "Tu dirección de correo electrónico", and "¿Te gusta el diseño de la aplicación?" with radio button options for "Sí" and "No".

Figura 5.3. Formulario Google

Los resultados obtenidos a través de la encuesta realizada han sido los que se presentan en forma de gráfica a continuación.

¿Te ha gustado el diseño de la aplicación?

Como se puede ver en la Figura 5.4. el diseño de la aplicación ha sido un éxito. En el proceso de desarrollo de la aplicación y gracias al asesoramiento de profesionales de la educación especial, en todo momento se ha pensado que fuese lo más sencillo y lo más intuitivo posible. Además, se ha querido incluir elementos como audios o el personaje que hagan que la aplicación sea un poco más amena y no sea una simple aplicación de ejercicios.

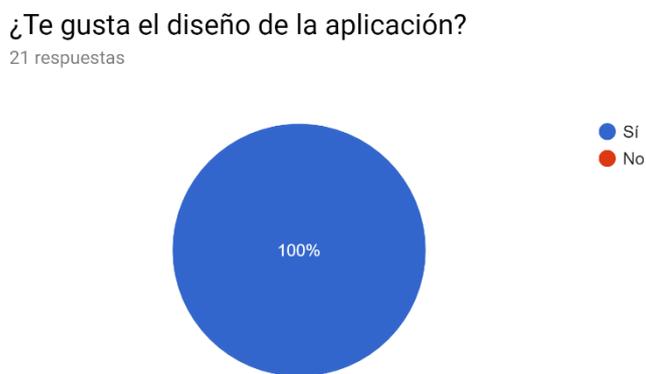


Figura 5.4. Resultados para la pregunta ¿Te ha gustado el diseño de la aplicación?

¿Has entendido lo que se pedía en cada juego?

Posiblemente esta cuestión no represente en gran medida la realidad ya que previamente, en la primera parte de la sesión, les había sido explicado en qué consistía cada juego. Lo más probable es que si la aplicación es subida a Play Store y es descargada por muchos usuarios, algunos de ellos no entiendan la totalidad de los juegos al 100%. En este caso como se puede observar en la Figura 5.5 lo han entendido cada juego la totalidad de los alumnos.

Por ello, se ha pensado que antes de comenzar cada juego, en futuras versiones, se va a incluir un tutorial donde se explique en qué consiste cada juego.

¿Has entendido lo que se pedía en cada juego?
21 respuestas

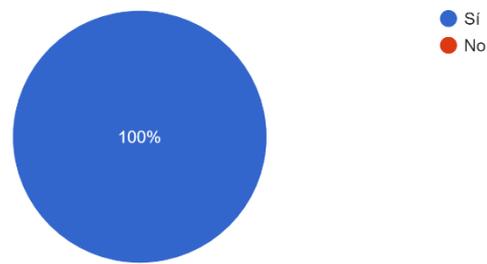


Figura 5.5. Resultados para la pregunta ¿Has entendido lo que se pedía en cada juego?

¿Te han resultado fáciles los juegos?

Los juegos diseñados en la aplicación han sido basados en la guía docente del programa Avanzas que fue prestada por la fundación Prodis. Con esta cuestión se pretende conocer el nivel de cada una de las clases donde se ha impartido la sesión para que en futuras versiones se adapte mejor. En la Figura 5.6 se ve que en casi todos los casos han resultado juegos sencillos.

¿Te han resultado fáciles los juegos?
21 respuestas

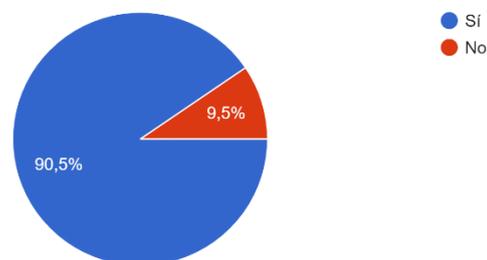


Figura 5.6. Resultados para la pregunta ¿Te han resultado fáciles los juegos?

¿Has necesitado la ayuda de algún profesor para utilizar la aplicación?

Como se ha comentado anteriormente, uno de los objetivos es que la aplicación sea lo más sencilla posible a la hora de operar con ella. Es por ello por lo que se ha querido conocer los casos en los que durante la sesión se ha necesitado ayuda de o bien algún profesor o bien ayuda mía. Como se muestra en la Figura 5.7 el 42,9% de los alumnos han necesitado ayuda, lo que corresponde a 9 de los 21 alumnos. Por tanto, gana aún más fuerza la idea de incluir breves tutoriales o indicaciones dentro de la aplicación.

¿Has necesitado la ayuda de algún profesor para utilizar la aplicación?

21 respuestas

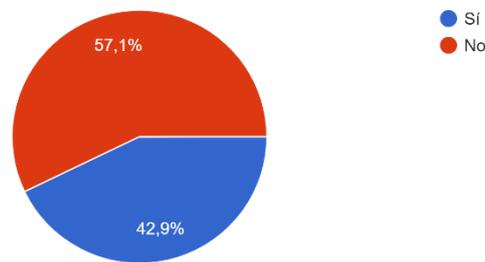


Figura 5.7. Resultados para la pregunta ¿Has necesitado la ayuda de algún profesor para utilizar la aplicación?

¿Crees que podrías utilizar la aplicación tú solo sin la ayuda de un profesor?

Con esta cuestión se pretende saber si después de la explicación y las indicaciones dadas a algunos de los alumnos para el uso de la aplicación, son capaces de utilizarla sin la ayuda de ningún profesor. En la Figura 5.8 se puede observar que la gran mayoría se serían capaces. Por tanto, los alumnos que en la cuestión anterior han respondido que han necesitado ayuda para operar con la aplicación, posiblemente la hayan necesitado por ser la primera vez que la utilizaban.

¿Crees que podrías utilizar la aplicación tú solo sin la ayuda de un profesor?

21 respuestas

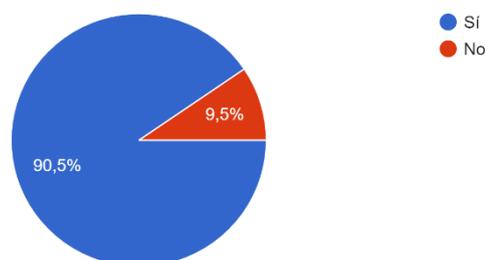


Figura 5.8. Resultados para la pregunta ¿Crees que podrías utilizar la aplicación tú solo sin la ayuda de un profesor?

¿Crees que podrías utilizar la aplicación en casa?

Además, se pretende que los alumnos de los programas que ayudan a las personas con discapacidad intelectual leve puedan hacer uso de la aplicación en casa y no solo en clase. La aplicación debe parecerles lo suficientemente sencilla y entretenida como para que puedan seguir practicando diversos aspectos en casa. Como se puede ver en la Figura 5.9 la mayoría de los alumnos después de las indicaciones de los profesores serían capaces de utilizar la aplicación en sus respectivas casas.

¿Crees que podrías utilizar la aplicación en casa?
21 respuestas

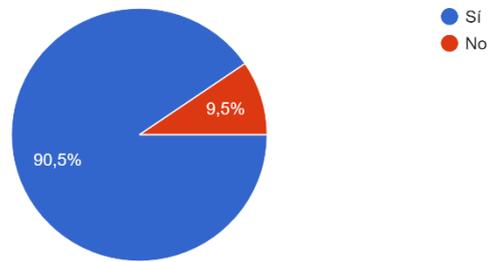


Figura 5.9. Resultados para la pregunta ¿Crees que podrías utilizar la aplicación en casa?

Al concluir la sesión, a cada uno de los alumnos se le pregunta de manera individual cuál ha sido su sección de juegos de la aplicación preferida obteniendo como resultado la gráfica de la Figura 5.10, en la que las Matemáticas han sido las más preferidas.

SECCIONES DE LA APLICACIÓN PREFERIDAS POR LOS ALUMNOS

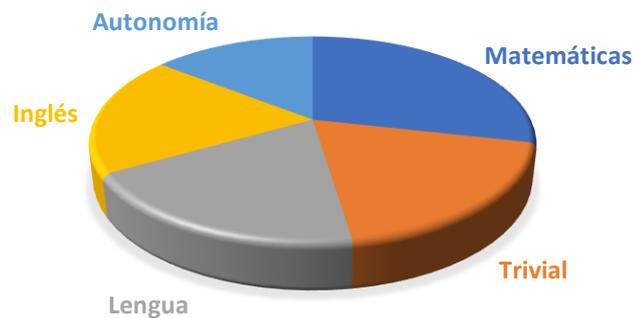


Figura 5.10. Secciones de la aplicación preferidas por los alumnos

Para finalizar la evaluación, por parte del profesorado de la Fundación Prodis, se han realizado cuatro recomendaciones que se han tenido en cuenta:

- 1) Los colores de la aplicación deben ser de una tonalidad más clara ya que algunos de los alumnos poseen problemas de visión y les cuesta distinguir la letra del fondo.
- 2) En la sección ‘Diver Frases’ de la aplicación, una vez que se ha arrastrado una palabra y ha sido colocada en el lugar correcto, eliminarla de las posibles palabras a arrastrar para no despistar usuario.
- 3) En la sección ‘Diver Acciones’ de la aplicación, una vez que se ha arrastrado un número y ha sido colocada en el lugar correcto, eliminarlo de los posibles números a arrastrar para no despistar usuario.
- 4) En la sección ‘Diver Trivial’ de la aplicación, si el usuario ha respondido de forma errónea a la pregunta, marcar de alguna manera la respuesta que es correcta.

Por otro lado, el profesorado del programa Avanzas de la Fundación Prodis ha redactado una carta plasmando las impresiones que tuvieron de la aplicación, la cual se adjunta en el Anexo A del presente documento. En ella, afirman que les gustó ver este tipo de aplicación con una interfaz adulta a la vez que sencilla y accesible.

Como se detalla en la sección 2.7 del documento y ahora reiteran los responsables de la Fundación Prodis, es bastante habitual que las aplicaciones cuyo principal objetivo es trabajar este tipo de contenidos, suelen tener un carácter demasiado infantil y por tanto para los chicos del rango de edad de 19 a 23 años con los que trabajan les parezca poco atractivo.

Además, haber explicado al profesorado que todas las puntuaciones de los alumnos almacenadas en Firebase podrían llegar a ser accesibles para tener un seguimiento desde otra aplicación ha generado contento.

Uno de los principales objetivos de la visita a Prodis era también escuchar nuevas ideas y sugerencias para las futuras versiones de la aplicación. En la carta redactada por parte de los responsables del programa Avanzas de la Fundación Prodis se han detallado algunas muy interesantes que se tendrán en cuenta.

Finalmente, se concluye dicha redacción afirmando que se trata de una aplicación novedosa cuyos contenidos están muy bien adaptados al nivel de sus alumnos además de tener algo diferente a las demás, su carácter adulto.



CAPÍTULO 6. Gestión del proyecto

6.1 Gestión del tiempo

En este capítulo de la memoria se va a exponer la planificación seguida para llevar a cabo el proyecto. Para ello se ha dividido en varias fases o tareas, tal y como se detalla en la Figura 1.3 de la sección 1.4. de las fases de desarrollo, a las que se les ha asignado un tiempo estimado de trabajo.

Un diagrama de Gantt es una herramienta para planificar en el tiempo todas aquellas actividades o tareas que componen un proyecto de larga duración [69]. En él se va a poder visualizar el comienzo y el final previstos de toda actividad y, de esta forma, se puede estar prevenido ante los posibles puntos críticos del proyecto [70]. Los puntos críticos de un proyecto son aquellos momentos en los que más de una actividad o tarea se lleva a cabo. Por otro lado, también facilita la localización de aquellas fases de desarrollo del proyecto donde se producirá inactividad, pudiendo calcular así los posibles retrasos de entrega del proyecto [70].

La Figura 6.1. muestra el diagrama de Gantt generado a través de Excel [71] en el que se ha marcado por colores cada una de las fases del proyecto (planificación en azul, ejecución en rojo, pruebas en verde y cierre en naranja).

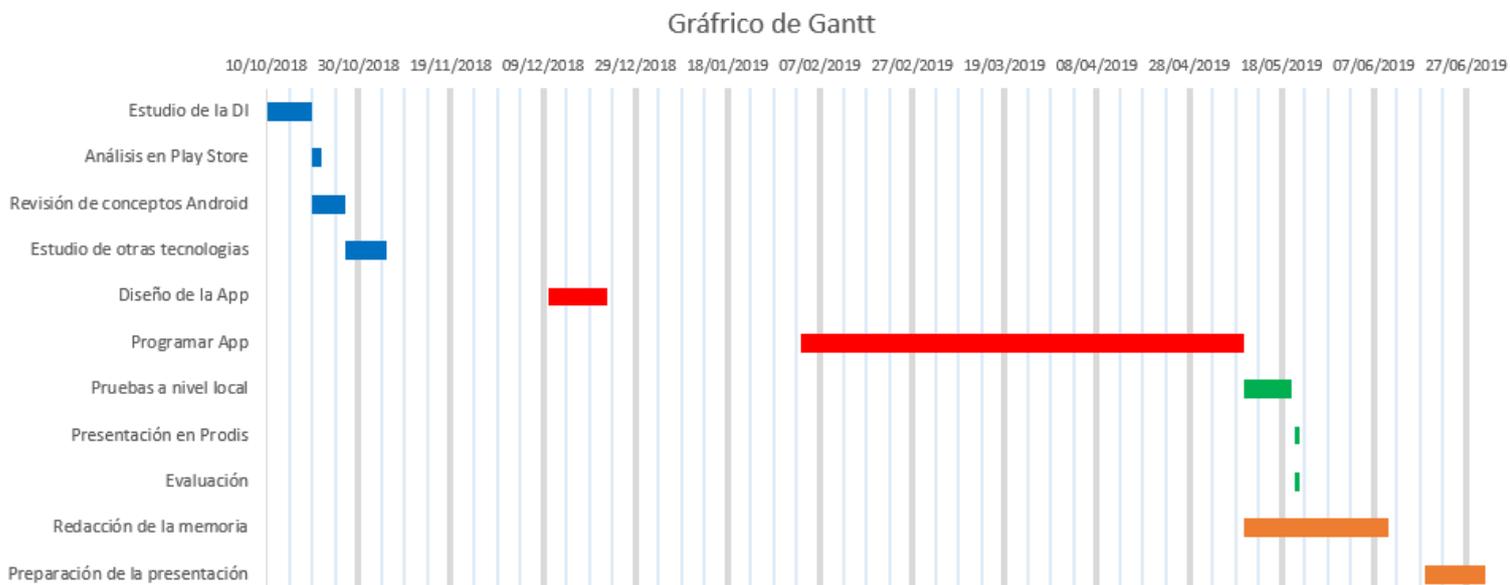


Figura 6.1. Diagrama de Gantt con la planificación temporal del proyecto

Además, en la Tabla 6.1. se muestran las fechas estimadas de inicio y de finalización de cada una de las actividades.

NOMBRE DE LA TAREA	FECHA INICIO	FECHA FIN	DURACIÓN (DÍAS)
Estudio de la DI	10/10/2018	20/10/2018	10
Análisis en Play Store	20/10/2018	22/10/2018	2
Revisión de conceptos Android	20/10/2018	27/10/2018	7
Estudio de otras tecnologías	27/10/2018	05/11/2018	9
Diseño de la App	10/12/2018	23/12/2018	13
Programar App	03/02/2019	10/05/2019	96
Pruebas a nivel local	10/05/2019	20/05/2019	10
Presentación en Prodis	21/05/2019	22/05/2019	1
Evaluación	21/05/2019	22/05/2019	1
Redacción de la memoria	10/05/2019	10/06/2019	31
Preparación de la presentación	18/06/2019	01/07/2019	13

Tabla 6.1. Fechas estimadas de inicio y fin del proyecto

Como se puede ver en la Figura 6.1 el proyecto se inició el 10 de octubre de 2018 y está previsto terminarlo el 1 de julio. En el primer mes de trabajo ha estado previsto llevar a cabo la etapa de planificación que incluye el estudio de la discapacidad intelectual, un análisis en Play Store de las aplicaciones que actualmente existen orientadas a la ayuda de personas con discapacidad intelectual, una revisión de algunos de los conceptos principales de Android que me han sido enseñados en la asignatura de Aplicaciones Móviles y un estudio de otras tecnologías que podrían ser utilizadas en la aplicación (MeaningCloud, sintetizador de texto a voz, etc.).

Además, han estado previstos dos periodos de inactividad debido al tiempo que se ha de dedicar al estudio de otras asignaturas del Grado para los exámenes. Uno de estos periodos es antes de realizar el diseño de la aplicación (del 5 de noviembre de 2018 al 11 de diciembre de 2018) y el segundo es después de llevar a cabo el diseño (del 23 de diciembre de 2018 al 3 de febrero de 2019). Por tanto, en total existen aproximadamente dos meses de inactividad.

Teniendo en cuenta que la fecha de entrega del proyecto es entre el 10 de junio de 2019 y el 16 de junio de 2019, a principios de febrero se ha programado el comienzo de la actividad que a priori más tiempo va a llevar, el desarrollo de la aplicación, para que en las últimas etapas de dicho desarrollo pueda alternarse las jornadas de trabajo con la redacción de la memoria y con las pruebas a nivel local. Por otro lado, para el día 21 de mayo de 2019 ha estado previsto la visita a la Fundación Prodis para llevar a cabo la evaluación de la aplicación, tal y como se detalla en el capítulo 5 del documento. Por tanto, el mes de mayo es un punto crítico del proyecto donde se debe concluir la programación de la aplicación, comenzar la redacción de la memoria,

realizar las pruebas a nivel local y en caso de que haya fallos corregirlos y preparar la sesión impartida en la Fundación Prodis el día 21 de mayo de 2019.

Está previsto terminar la redacción de la memoria aproximadamente el 10 de junio para poder entregarlo a tiempo.

Se puede observar que existe un nuevo periodo de inactividad entre el 10 y el 18 de junio de 2019. Esto se debe a que para el día 17 del mismo mes el autor del proyecto tiene programado un examen universitario y, por tanto, se invertirá tiempo en el estudio. Por último, después de dicho examen, se concluirá el proyecto con la preparación de la presentación que se llevará a cabo ante un tribunal de la Universidad Carlos III el día 4 de julio de 2019.

6.2 Presupuesto estimado

En esta sección de la memoria se procede a realizar una estimación del presupuesto que se debería invertir para llevar a cabo este plan. Para ello, hay que distinguir entre los dos principales tipos de costes en un proyecto: costes directos y costes indirectos [72]:

- Los costes directos son todos aquellos que tienen relación con los materiales empleados para la realización del proyecto y con la mano de obra, es decir, los salarios de los empleados.
- Por otro lado, los costes indirectos son los que suponen todo aquello que hace posible el proyecto, como por ejemplo es el alquiler de un local, la luz, el agua, etc.

Una vez que se han definido las actividades a realizar en el proyecto, a continuación, en la Tabla 6.2. se va a realizar una lista con los principales materiales que son necesarios en la ejecución de todos los procesos, su precio, su tiempo de vida o tiempo de amortización, su coste por mes o coste de amortización y el coste que supone en el proyecto desarrollado. Para calcular el coste de amortización se ha dividido el precio entre el tiempo de vida en meses y para el coste en el proyecto, se ha multiplicado por los 9 meses que dura el proyecto:

Material	Precio	Tiempo de vida	Coste de amortización	Total
Ordenador portátil ASUS Transformer Book Flip TP301UA (I5)	400 €	60 meses (5 años)	6,6 €/mes	59,4 €
Ratón inalámbrico Microsoft	6,75 €	60 meses (5 años)	0,1125 €/mes	1,0125 €

Cable USB	3 €	60 meses (5 años)	0,05 €/mes	0,45 €
Ordenador Mac 9.1	1200 €	72 meses (6 años)	16,66 €/mes	149,94 €
Huawei P10 Lite	200 €	36 meses	5,55 €/mes	49,95 €
Huawei P20 Lite	200 €	36 meses	5 €/mes	45 €
BQ Aquaris M5	200 €	24 meses	8,33 €/mes	74,97 €
BQ Aquaris M4.5	180 €	24 meses	7,5 €/mes	67,5 €
Microsoft Office 365	-	-	8,80 €/mes	79,2 €

Tabla 6.2. Costes directos. Costes de material

Seguidamente, se ha de calcular los gastos producidos por el pago salarial de los empleados. Se va a tomar como referencia salarial de un ingeniero de telecomunicaciones la cifra que se indica en [73]. En la Tabla 6.3. se ha calculado el coste por hora del desarrollador, se indican los días que se han trabajado y a partir de ello se han calculado las horas de trabajo partiendo de unas 5 horas diarias aproximadamente. Finalmente, multiplicando las horas trabajadas por el coste del empleado por hora, se ha obtenido un coste total.

Nombre	Salario bruto mensual	Coste por día	Coste por hora	Días	Horas(5h/día)	Total
Daniel Ruiz García	2425 €	78,226 €	3,26 €	193	965	3145,9 €

Tabla 6.3. Costes directos. Costes de personal

Finalmente, para calcular el coste total aproximado de llevar a cabo este proyecto, hay que tener en cuenta los gastos indirectos. En este caso se va a tener en cuenta alquiler de local donde se ha realizado el proyecto, la luz, el agua y el internet. En la Tabla 6.4. se muestra en forma de tabla los gastos indirectos:

	Precio mensual	Total
Alquiler	350 €	3150 €
Luz	35 €	315 €
Agua	35 €	315 €
Internet	25 €	225 €

Tabla 6.4. Costes indirectos



Por tanto, sumando todos los costes totales de las tablas se ha obtenido un precio estimado del proyecto de SIETE MIL SEISCIENTOS SETENTA Y OCHO CON TREINTA Y DOS EUROS.

6.3 Marco regulador

6.3.1 Políticas de desarrolladores de Google Play

A la hora de desarrollar una aplicación móvil se debe ser consciente de la responsabilidad que conlleva cumplir ciertos aspectos legales, algo que en muchos casos no ocurre. De esta manera, el programador evitará ser sancionado. A continuación, se muestran las principales políticas relacionadas con el proyecto a las que el desarrollador de la aplicación móvil ha debido ser fiel para que en el caso de subirla a Google Play no sea retirada [85]:

- **Contenido restringido:** toda aquella aplicación dirigida para niños, pero cuyo contenido sea para adultos será retirada. Además, aplicaciones que inciten al odio, la violencia, al acoso, que fomenten la venta de productos peligrosos o que contengan contenido sexual explícito no serán admitidas por Google Play. Por otro lado, las herramientas financieras engañosas han ido vetadas, a diferencia de los anuncios y contenido de juegos de apuestas, los cuales son permitidos siempre y cuando cumplan los requisitos que se detallan en [74]. También está prohibida la facilitación de actividades ilegales como la venta de drogas o la de alcohol a menores. Finalmente, en [75] se detalla una lista de las sustancias que no han sido aprobadas y que por lo tanto no está permitida su promoción ni venta.
- **Suplantación de identidad:** está prohibido el uso del nombre de otras marcas, del logotipo de otra aplicación o incluso el nombre de otra aplicación. Por otro lado, es importante tener en cuenta la ley de propiedad intelectual, la cual se detalla en la sección 7.2.
- **Privacidad:** en el caso de querer utilizar los datos del usuario, se debe notificar. Además, en el caso de que sea necesario el acceso a determinados servicios por parte de la aplicación, se deben pedir los permisos necesarios al usuario. Mientras la aplicación no haya obtenido las autorizaciones pertinentes, no puede interferir en el dispositivo del usuario, ni en ordenadores, redes, servicios o APIs. De forma evidente, Google Play tampoco admite aplicaciones cuyos fines sean dañar a los usuarios, les vigilen o les roben información. Por último, el contenido de la aplicación debe ser claro y preciso. Es decir, que el usuario sea conocedor a la perfección de las funciones que implementa la aplicación. De esta manera, se evita el comportamiento engañoso. Las condiciones descritas se suman a la ley de protección de datos, la cual se procede a detallar en a la sección 7.2.
- **Obtención de ingresos:** toda aquella aplicación cuyo principal fin sea la venta o compra de objetos, deben ser fieles a las normas que se detallan en [76].



Además, si el usuario de una aplicación cancela una suscripción de pago, seguirá recibiendo el contenido de la suscripción hasta que concluya ésta. La publicidad engañosa está vetada y en [77] se muestran las directrices que se han de seguir para la publicación de anuncios de juegos de apuestas.

- Spam: queda vetado por parte de Google Play.
- Funcionalidad mínima: la aplicación debe ofrecer una experiencia mínima a los usuarios.
- Cumplimiento obligatorio: las políticas descritas deben cumplirse tanto en el contenido que se ofrece en la aplicación como en el contenido al que se accede desde la misma incluidos los anuncios. Google Play tras comprobar que la aplicación no es apta para estar en el mercado, la retira y por medio de un correo procede a la notificación del motivo.

6.3.2 Legislación

Las principales medidas legales tomadas para llevar a cabo el desarrollo de la aplicación móvil del Trabajo Fin de Grado son:

- Ley orgánica 3/2018 de Protección de Datos Personales y garantía de los derechos digitales (5 de diciembre). Según el artículo 1 de dicha ley [78]:

“El derecho fundamental de las personas físicas a la protección de datos personales, amparado por el artículo 18.4 de la Constitución, se ejercerá con arreglo a lo establecido en el Reglamento (UE) 2016/679 y en esta ley orgánica.”

Si se accede al artículo 2 del Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo se observa [79]:

“El presente Reglamento se aplica al tratamiento total o parcialmente automatizado de datos personales, así como al tratamiento no automatizado de datos personales contenidos o destinados a ser incluidos en un fichero.”

En la aplicación desarrollada, cuando los usuarios llevan a cabo el registro, se obtiene su correo electrónico. Sin embargo, en el artículo 6.1. de dicho reglamento se puede leer [79]:

“El tratamiento solo será lícito si se cumple al menos una de las siguientes condiciones:

a) el interesado dio su consentimiento para el tratamiento de sus datos personales para uno o varios fines específicos.

b) el tratamiento es necesario para la ejecución de un contrato en el que el interesado es parte o para la aplicación a petición de este de medidas precontractuales.



c) el tratamiento es necesario para el cumplimiento de una obligación legal aplicable al responsable del tratamiento.

d) el tratamiento es necesario para proteger intereses vitales del interesado o de otra persona física.

e) el tratamiento es necesario para el cumplimiento de una misión realizada en interés público o en el ejercicio de poderes públicos conferidos al responsable del tratamiento.

f) el tratamiento es necesario para la satisfacción de intereses legítimos perseguidos por el responsable del tratamiento o por un tercero, siempre que sobre dichos intereses no prevalezcan los intereses o los derechos y libertades fundamentales del interesado que requieran la protección de datos personales, en particular cuando el interesado sea un niño.”

Por tanto, teniendo en cuenta el artículo 6.1.a se ha decidido antes de llevar a cabo el registro de los usuarios en la aplicación, solicitar el consentimiento del uso de sus datos, como es el correo electrónico, mediante una ventana donde deba aceptar las políticas de la aplicación. El mensaje que aparece en dicha ventana es:

“De conformidad con el vigente Reglamento (UE) 2016/679, se informa que los datos de carácter personal de los Usuarios de la aplicación se tratarán única y exclusivamente para llevar a cabo el registro de este en la plataforma encargada de la autenticación de usuarios. Dicho tratamiento de sus datos estará amparado en su propio consentimiento. Al pulsar el botón “LOGIN”, el Usuario consiente el tratamiento de sus datos por parte del desarrollador de la aplicación.

Asimismo, le informamos que, salvo obligación legal o consentimiento expreso por su parte, no cederemos sus datos a terceros.

Igualmente, se informa al Usuario que en cualquier momento puede ejercitar los derechos de acceso, rectificación o supresión de datos, así como disponer de otros derechos regulados en el Reglamento (UE) 2016/679, notificándolo a DANIEL RUIZ GARCÍA, 1003159000@alumnos.uc3m.es.”

- Ley de Propiedad Intelectual. Según el artículo 1 del Real Decreto Legislativo 1/1996 [80]:

“La propiedad intelectual de una obra literaria, artística o científica corresponde al autor por el solo hecho de su creación.”

En la aplicación desarrollada para el Trabajo Fin de Grado todas las imágenes y audios incluidos están libres de uso y no es necesario el reconocimiento del autor.

- Ley 9/2014 General de Telecomunicaciones (9 de mayo). Según el artículo 43 [81]:



“Cualquier tipo de información que se transmita por redes de comunicaciones electrónicas podrá ser protegida mediante procedimientos de cifrado.”

A la hora de utilizar el sistema de autenticación de la aplicación desarrollada, cada contraseña que se envía a la plataforma de Firebase es cifrada por el sistema.



CAPÍTULO 7. Conclusiones y trabajos futuros

En este capítulo se lleva a cabo una recopilación de todas las tareas realizadas. A modo de conclusión del proyecto, se expone un balance del trabajo realizado durante los últimos meses. Además, se muestran las posibles futuras líneas de trabajo de la aplicación desarrollada.

7.1 Conclusiones

En este Trabajo Fin de Grado se ha desarrollado una aplicación móvil para dispositivos móviles Android cuyo principal objetivo es favorecer la autonomía de personas con discapacidad intelectual leve trabajando el área lingüística, de autogestión, cultural y matemática.

Al iniciar la aplicación, la persona que maneja el dispositivo móvil se encuentra con un sistema de autenticación en el que, en el caso de no estar registrado en el sistema, deberá hacerlo introduciendo un correo electrónico y una contraseña. Si el usuario ya está registrado en el sistema, simplemente debe introducir su correo electrónico y contraseña y pulsar el botón de 'Login'.

Tras haberse identificado el usuario, aparece el menú principal de la aplicación donde hay 5 botones: el botón superior izquierdo destinado al área lingüística, el botón superior derecho destinado a las matemáticas, el botón inferior izquierdo destinado al área cultura, el botón inferior derecho destinado a la lengua inglesa y el botón central que se ha destinado a actividades de autogestión.

Pulsando en el botón superior izquierdo se accede a una lista de juegos del área lingüística con sus respectivas barras de progreso que se actualizan en función del nivel de actividad del usuario en un determinado juego. En la actual versión de la aplicación existen 3 juegos distintos. En uno de ellos el usuario debe ordenar las palabras que aparecen en pantalla para formar frases. Por otro lado, el segundo juego es orientado a la comprensión lectora y en él aparecerá un texto sobre el que se realizarán preguntas. Finalmente, en el tercer juego el usuario practicará la retención de ideas o memoria. En él aparecerán 5 imágenes de forma sucesiva y posteriormente el jugador deberá seleccionar las palabras que estén relacionadas con dichas imágenes.

Pulsando el botón superior derecho del menú principal de la aplicación se tiene acceso a una lista de juegos con los que el usuario será capaz de practicar las matemáticas y el manejo del euro. Cada juego, además, tiene su respectiva barra de progreso la cual se actualiza en función del nivel de actividad del usuario. En dicha lista se cuenta con 6 juegos. En el primero de ellos aparece una operación matemática (sumas y restas) sin resultado y el jugador debe seleccionar la respuesta correcta de entre las 5 que se le ofrece, las cuales aparecen y desaparecen en distintos lugares de la pantalla. El segundo de los juegos consiste en preguntas tipo test sobre fracciones en las que el usuario deberá escoger una de entre las 3 respuestas posibles. El tercer



juego tiene la misma dinámica que el primero, es decir, se muestra una operación matemática con fracciones (sumas y restas) sin resultado y el jugador debe elegir una de entre las 5 respuestas posibles que se le ofrecen, las cuales aparecen y desaparecen en distintas posiciones de la pantalla. En el cuarto juego, también se practican las fracciones y en él al jugador se le pregunta por la fracción que representa una imagen y tendrá que rellenar dos casillas y pulsar el botón de confirmar para comprobar si su respuesta es correcta. Por último, esta sección de la aplicación cuenta con dos juegos mediante los cuales el usuario puede practicar el manejo del euro. En el primero de ellos, se le asigna un precio a un objeto y el jugador debe arrastrar las monedas y billetes necesarios para comprarlo. Por otro lado, el segundo juego consiste en confirmar si las cantidades de dinero representadas con billetes y monedas en dos imágenes distintas son las mismas o no.

A través del botón inferior izquierdo de la pantalla principal de la aplicación se accede al área de trabajo cultural. En dicha sección aparece una ruleta con distintos temas sobre los que se realizarán preguntas: deportes, ciencia, entretenimiento, literatura, historia y geografía. Tras pulsar en la ruleta comienza a girar y de forma aleatoria se selecciona uno de los temas. Al usuario se le realiza una pregunta tipo test del tema y deberá escoger una de entre las tres respuestas posibles.

Desde el botón inferior derecho del menú principal de la aplicación, el usuario puede acceder a la sección donde se lleva cabo la práctica del inglés. Solamente hay una actividad la cual consiste en el tradicional juego de memoria de levantar cartas y relacionar imágenes con la palabra que la define escrita en inglés (juego memory).

Por último, pulsando en el botón central del menú principal de la aplicación se tiene acceso a la sección orientada a la autogestión. En ella aparece un segundo menú con tres botones habilitados. En el primero de ellos el usuario accede al lugar desde donde se trabaja la orientación temporal. En él se puede observar una lista con las diferentes juegos disponibles y sus respectivas barras de progreso. Actualmente hay dos actividades: en el primero de ellos el jugador debe unir horas representadas en un reloj analógico con horas representadas en un reloj digital y en el segundo de ellos el usuario debe contestar preguntas tipo test con tres respuestas posibles sobre la hora representada en una imagen. En el segundo botón habilitado en el menú de la sección de autogestión se puede practicar la toma de decisiones. En esta actividad al usuario se le introduce una situación específica y él deberá arrastrar al personaje o mascota de la aplicación de entre varias respuestas posibles la que mejor se adecúa. Finalmente, el tercer y último botón disponible en la sección de autogestión el usuario debe ordenar las tareas que son necesarias para llevar a cabo una actividad.

Además, en el menú principal cada cierto tiempo se abre una conversación con el personaje de la aplicación en la cual se le pregunta usuario su opinión acerca de la aplicación desarrollada o, de forma más específica, acerca de algún juego concreto. El texto de la respuesta del usuario es examinado por el analizador de sentimiento de MeaningCloud y a través del resultado que ofrece, el personaje termina la conversación con una contestación u otra.



Asimismo, cada una de las opiniones y puntuaciones de los usuarios registrados en el sistema, serán almacenados en la base de datos en tiempo real de Firebase.

Para llevar a cabo el desarrollo de la aplicación ha sido necesario realizar un estudio sobre la discapacidad intelectual e informarse de los programas de educación existentes para ellos, además de atender a los consejos de una profesional de la educación especial, los cuales me han sido de gran utilidad. Por otro lado, el plan docente del programa Avanzas prestadas por la Fundación Prodis ha resultado de mucho provecho ya que ha sido utilizado de guía a la hora de distinguir las diferentes secciones de la aplicación.

Se ha realizado un estudio del sistema operativo Android con el objetivo de informarse de su origen, las versiones existentes, su arquitectura y algunas de sus principales características como es el sistema de compilación Gradle que posee.

Al igual que de Android, también se ha ejecutado un estudio del software Android Studio obteniendo información de sus ventajas con respecto a otros entornos de programación Android y de algunas de sus principales herramientas, como es el emulador, y de sus principales características, como es su interfaz.

Anteriormente no había sido utilizado por el desarrollador ningún sistema de autenticación de usuarios y se ha llevado a cabo un estudio del sistema de autenticación que ofrece Firebase entre sus múltiples servicios.

La aplicación desarrollada debía contar con un gran módulo de almacenamiento. Por un lado, se debe almacenar en algún lugar todas las preguntas, respuestas posibles y respuestas correctas. Por otro lado, opiniones y puntuaciones de los usuarios se deben guardar también. Por ello, para el primer caso se ha realizado un estudio del motor de gestión de datos SQLite y para el segundo uno de las bases de datos en tiempo real de Firebase.

A modo de conclusión general, hay que decir que los objetivos marcados al principio del documento han sido mayoritariamente cumplidos y se ha desarrollado una aplicación que es diferente al resto de aplicaciones que trabajan este tipo de aspectos. Además, tras recibir la carta de impresión de la aplicación de la Fundación Prodis, ha demostrado al programador que es efectiva y su contenido se adecua bien a las personas con discapacidad intelectual leve.

7.2 Líneas de trabajo futuro

Algunas de las ideas del desarrollador no se han podido implementar en la aplicación por falta de tiempo. Sin embargo, serán tenidas en cuenta para las futuras versiones. Por otro lado, en la visita a la Fundación Prodis, los responsables del programa Avanzas, han hecho algunas recomendaciones para que la aplicación mejore en algunos aspectos. Dichas ideas y recomendaciones se detallan a continuación:

- Implementar de forma completa un menú deslizante en el menú principal de la aplicación donde el usuario pueda observar sus resultados en las actividades,



personalizar la interfaz de la aplicación a su gusto, visualizar una serie de tutoriales de los juegos y compartir la aplicación.

- Cambiar los colores de la aplicación y tamaño de fuente. Deben ser colores menos saturados y el tamaño de fuente debe ser más grande ya que algunas personas con discapacidad intelectual tienen problemas de visión y tienden a acercarse mucho a la pantalla del dispositivo móvil.
- Implementar nuevos juegos o actividades tras haberse seguido informando sobre los distintos métodos de aprendizaje que se aplican en la educación especial de personas con discapacidad intelectual.
- Implementar juegos para grupos.
- Más a largo plazo, se podrían implementar juegos orientados para cada uno de los niveles de discapacidad intelectual explicados en la sección 2.7.1.: leve, moderado, grave y profundo.
- Introducir en la aplicación un sistema de reconocimiento de voz, el cual el usuario pueda activar o desactivar desde el menú deslizante ya que no a todas las personas con discapacidad intelectual leve se las entiende con total claridad.
- Además de contar con un sistema de autenticación donde el usuario deba introducir su correo electrónico y contraseña, implementar un sistema donde el usuario pueda acceder a su cuenta a través del reconocimiento de la huella de uno de sus dedos.
- Incluir versiones en diversos idiomas como el alemán y el inglés.
- Introducir un sistema de interacción por voz que se pueda activar y desactivar, ya que era uno de los objetivos definidos al principio del documento.
- Simplificar el código.
- Mas a largo plazo, gracias a que todas las puntuaciones son almacenadas en la base de datos en tiempo real de Firebase, se podrá crear una nueva aplicación que acceda a esa información y desvele a los responsables de una clase específica la evolución de sus alumnos.



BIBLIOGRAFÍA

- [1] Asociación Americana de Psiquiatría (2014). *Manual Diagnóstico y Estadístico de los Trastornos Mentales (DSM-5)* (5ª edición). Madrid: Editorial Médica Panamericana. [En línea]. Disponible en:
<http://www.codajic.org/sites/www.codajic.org/files/DSM%205%20%20Novedades%20y%20Criterios%20Diagn%C3%B3sticos.pdf>
- [2] Ministerio de sanidad, consumo y bienestar, «Base estatal de datos de personas con valoración del grado de discapacidad» IMSERSO. Madrid,2018. [En línea]. Disponible en:
http://imserso.es/InterPresent2/groups/imserso/documents/binario/bdepcd_2016.pdf
- [3] Plena Inclusión, «¿Qué es discapacidad intelectual?» [En línea] Disponible en <https://www.plenainclusion.org/discapacidad-intelectual/que-es-discapacidad-intelectual>
- [4] I. García Alonso, Colegio Oficial de Psicólogos de Madrid (2005), *Intervención Psicosocial* (14ª edición) «Concepto actual de la discapacidad» [En línea] Disponible en <http://www.copmadrid.org/webcopm/publicaciones/social/100008.pdf>
- [5] Fundación Prodis, sitio web [En línea] disponible en <https://www.fundacionprodis.org/>
- [6] N. Benbourahla. *Android 7. Principios del desarrollo de aplicaciones Java*. Barcelona: Recursos informáticos, (2017)
- [7] Fundación Prodis, «Rasgos de identidad» [En línea] Disponible en <https://www.fundacionprodis.org/fundacion-prodis/rasgos-de-identidad/>
- [8] Wikipedia, «Open Handset Alliance» [En línea] Disponible en https://es.wikipedia.org/wiki/Open_Handset_Alliance
- [9] Wikipedia, «Android» [En línea] Disponible en <https://es.wikipedia.org/wiki/Android>
- [10] Android Developers, «Paneles de control » [En línea] Disponible en <https://developer.android.com/about/dashboards?hl=es-419>
- [11] Wikipedia, «Linux» [En línea] Disponible en https://es.wikipedia.org/wiki/N%C3%BAcleo_Linux
- [12] Wikipedia, «Código abierto» [En línea] Disponible en https://es.wikipedia.org/wiki/Software_de_c%C3%B3digo_abierto
- [13] Wikipedia, «Software libre» [En línea] Disponible en https://es.wikipedia.org/wiki/Software_libre
- [14] XatakaAndroid, «¿Qué es Android?» [En línea] Disponible en <https://www.xatakandroid.com/sistema-operativo/que-es-android>
- [15] TuProgramación, «¿Qué es Dalvik?» [En línea] Disponible en <http://www.tuprogramacion.com/glosario/que-es-dalvik/>
- [16] Wikipedia, «Recolector de basura» [En línea] Disponible en https://es.wikipedia.org/wiki/Recolector_de_basura



- [17] Android Developers, « Arquitectura de la plataforma» [En línea] Disponible en <https://developer.android.com/guide/platform?hl=es>
- [18] Software e Comunicaciones UC3M, « Arquitectura de Android» [En línea] Disponible en <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>
- [19] Oracle, « Java SE Development Kit 8 Downloads» [En línea] Disponible en <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- [20] Wikipedia, « IntelliJ IDEA» [En línea] Disponible en https://es.wikipedia.org/wiki/IntelliJ_IDEA
- [21] AndroidStudioFaqs, «¿Qué es gradle en Android Studio?» [En línea] Disponible en <https://androidstudiofaqs.com/conceptos/que-es-gradle-en-android-studio>
- [22] Wikipedia, « Máquina virtual Java» [En línea] Disponible en https://es.wikipedia.org/wiki/M%C3%A1quina_virtual_Java
- [23] Android Developers, « Conoce Android Studio» [En línea] Disponible en <https://developer.android.com/studio/intro/?hl=ES>
- [24] Wikipedia, « Google App Engine» [En línea] Disponible en https://es.wikipedia.org/wiki/Google_App_Engine
- [25] Android Developers, « Como actualizar el IDE y las herramientas del SDK» [En línea] Disponible en <https://developer.android.com/studio/intro/update>
- [26] Android Developers, « Funciones de Android Studio» [En línea] Disponible en <https://developer.android.com/studio/features?hl=es>
- [27] Wikipedia, « Java Native Interface» [En línea] Disponible en https://es.wikipedia.org/wiki/Java_Native_Interface
- [28] SQLite, « What is SQLite? » [En línea] Disponible en <https://www.sqlite.org/index.html>
- [29] SQLite, « Features of SQLite » [En línea] Disponible en <https://www.sqlite.org/features.html>
- [30] Wikipedia, « Interfaz de línea de comandos » [En línea] Disponible en https://es.wikipedia.org/wiki/Interfaz_de_l%C3%ADnea_de_comandos
- [31] Hermosa Programación « Tutorial de bases dedatos SQLite en aplicaciones Android » [En línea] Disponible en <http://www.hermosaprogramacion.com/2014/10/android-sqlite-bases-de-datos/>
- [32] Android Developers, « FragmentManager, *beginTransaction()* » [En línea] Disponible en [https://developer.android.com/reference/android/app/FragmentManager#beginTransaction\(\)](https://developer.android.com/reference/android/app/FragmentManager#beginTransaction())
- [33] MeaningCloud, « Analítica de texto » [En línea] Disponible en <https://www.meaningcloud.com/es/soluciones/analitica-de-texto>
- [34] MeaningCloud, « APIs » [En línea] Disponible en <https://www.meaningcloud.com/developer/apis>



- [35] Wikipedia, « Firebase » [En línea] Disponible en <https://es.wikipedia.org/wiki/Firebase>
- [36] Firebase, « Servicios Firebase » [En línea] Disponible en <https://firebase.google.com/products>
- [37] Wikipedia, « NoSQL » [En línea] Disponible en <https://es.wikipedia.org/wiki/NoSQL>
- [38] Firebase, « Agrega Firebase al proyecto de Android » [En línea] Disponible en <https://firebase.google.com/docs/android/setup?hl=es-419>
- [39] Firebase, « Consola Firebase » [En línea] Disponible en <https://console.firebase.google.com/u/0/>
- [40] Firebase, « Bibliotecas disponibles » [En línea] Disponible en <https://firebase.google.com/docs/android/setup#available-libraries>
- [41] Universidad Autónoma de Barcelona, « La conversión de texto en habla » [En línea] Disponible en http://liceu.uab.cat/~joaquim/speech_technology/tecnol_parla/synthesis/tts_general/conversion_texto_habla.html
- [42] Xatakamovil, « Así es como Android se ha comido el mercado en diez años » [En línea] Disponible en <https://www.xatakamovil.com/sistemas-operativos/asi-como-android-se-ha-comido-mercado-diez-anos>
- [43] Datawrapper, sitio web de la herramienta disponible en <https://www.datawrapper.de/>
- [44] AppleSfera, « Lo ingresos de la App Store prácticamente han doblado a los de Google Play en los que llevamos de año » [En línea] Disponible en <https://www.applesfera.com/app-store-1/ingresos-app-store-practicamente-han-doblado-google-play-que-llevamos-ano>
- [45] Android Developers, « An update on Eclipse Android Developer Tools » [En línea] Disponible en <https://android-developers.googleblog.com/2015/06/an-update-on-eclipse-android-developer.html>
- [46] E. Guevara Cuesta, “Aproximación conceptual a la discapacidad intelectual”, en *Alumnado con discapacidad cognitiva*. (2019)
- [47] Wikipedia, « Anoxia » [En línea] Disponible en <https://es.wikipedia.org/wiki/Anoxia>
- [48] Mayo Clinic, « Enfermedad desmielizante ¿Qué puedes hacer? » [En línea] Disponible en <https://www.mayoclinic.org/es-es/diseases-conditions/multiple-sclerosis/expert-answers/demyelinating-disease/faq-20058521>
- [49] Mayo Clinic, « Placenta previa » [En línea] Disponible en <https://www.mayoclinic.org/es-es/diseases-conditions/placenta-previa/symptoms-causes/syc-20352768>
- [50] Wikipedia, « Privación sensorial » [En línea] Disponible en https://es.wikipedia.org/wiki/Privaci%C3%B3n_sensorial
- [51] Hospital general Universitario Gregorio Marañón, «Retraso ponderal » [En línea] Disponible en <http://www.madrid.org/cs/Satellite?blobcol=urldata&blobheader=application%2Fpdf>



[f&blobheadername1=Content-disposition&blobheadername2=cadena&blobheadervalue1=filename%3DRETRASO+PONDERAL_13_07_2016.pdf&blobheadervalue2=language%3Des%26site%3DHospitalGregorioMaranon&blobkey=id&blobtable=MungoBlobs&blobwhere=1352912803766&ssbinary=true](https://iacapap.org/wp-content/uploads/C.1-Discapacidad-Intelectual-SPANISH-2018.pdf)

[52] X. Ke & J. Liu. “Capítulo 1. Discapacidad intelectual”, en *Trastornos del desarrollo*. China: M. Irarrázaval & A. Martín, 2017, 1-28 [En línea]. Disponible en: <https://iacapap.org/wp-content/uploads/C.1-Discapacidad-Intelectual-SPANISH-2018.pdf>

[53] MedlinePlus, « Síndrome X frágil » [En línea] Disponible en <https://medlineplus.gov/spanish/fragilexsyndrome.html>

[54] Wikipedia, « Fenilcetonuria » [En línea] Disponible en <https://es.wikipedia.org/wiki/Fenilcetonuria>

[55] Mayo Clinic, « Ataxia » [En línea] Disponible en <https://www.mayoclinic.org/es-es/diseases-conditions/ataxia/symptoms-causes/syc-20355652>

[56] Wikipedia, « Galactosemia » [En línea] Disponible en <https://es.wikipedia.org/wiki/Galactosemia>

[57] Ministerio de asuntos exteriores, unión europea y cooperación, « Organización Mundial de la Salud (OMS) » [En línea] Disponible en <http://www.exteriores.gob.es/RepresentacionesPermanentes/OficinadelasNacionesUnidas/es/quees2/Paginas/Organismos%20Especializados/OMS.aspx>

[58] Asociación Americana de Psiquiatría (2000). *Manual Diagnóstico y Estadístico de los Trastornos Mentales (DSM-IV)* (4ª edición). Madrid: Editorial Médica Panamericana. [En línea]. Disponible en: <http://www.mdp.edu.ar/psicologia/psico/cendoc/archivos/Dsm-IV.Castellano.1995.pdf>

[59] Ministerio de educación y formación profesional, « Discapacidad intelectual: media, severa y profunda. ¿Cuáles son las causas? » [En línea] Disponible en http://descargas.pntic.mec.es/cedec/atencion_diver/contenidos/nee/discapacidadmediaseveraprofunda/cules_son_las_causas.html

[60] Google Play, « Terapia de Lenguaje y Cognitiva con MITA » [En línea] Disponible en <https://play.google.com/store/apps/details?id=com.imagination.mita&gl=ES>

[61] Google Play, « PictoTEA » [En línea] Disponible en <https://play.google.com/store/apps/details?id=ar.com.velociteam.pictoTEA&gl=ES>

[62] Google Play, « iSECUENCIAS LITE » [En línea] Disponible en <https://play.google.com/store/apps/details?id=air.com.fundacionplanetaimaginario.isequencesliteversion&gl=ES>

[63] Google Play, « Conciencia fonológica - Juegos para autismo » [En línea] Disponible en <https://play.google.com/store/apps/details?id=com.meza.conciencia.fonologica&gl=ES>



- [64] Google Play, << Soy Cappaz>> [En línea] Disponible en <https://play.google.com/store/apps/details?id=com.mapfre.soycappaz&gl=ES>
- [65] Understood, << ¿El autismo es una discapacidad del aprendizaje>> [En línea] Disponible en <https://www.understood.org/es-mx/learning-attention-issues/getting-started/what-you-need-to-know/is-autism-a-learning-disability>
- [66] Android Developers, << Android Studio>> [En línea] Disponible en <https://developer.android.com/studio/index.html?hl=es-419>
- [67] MeaningCloud, sitio web [En línea] Disponible en <https://www.meaningcloud.com/es>
- [68] Fundación Prodis, << Programa Avanzas>> [En línea] Disponible en <https://www.fundacionprodis.org/programas-servicios/programa-avanzas/>
- [69] OBS Business School, << Organizar el cronograma de actividades con el Diagrama de Gantt>> [En línea] Disponible en <https://www.obs-edu.com/es/blog-project-management/diagramas-de-gantt/organizar-el-cronograma-de-actividades-con-el-diagrama-de-gantt>
- [70] OBS Business School, << ¿Qué es un diagrama de Gantt y para qué sirve?>> [En línea] Disponible en <https://www.obs-edu.com/es/blog-project-management/diagramas-de-gantt/que-es-un-diagrama-de-gantt-y-para-que-sirve>
- [71] Saber programas. *Excel-Cómo hacer un diagrama de Gantt o cronograma utilizando los gráficos. Tutorial en español HD*. (31 de julio de 2017) [Vídeo en línea] Disponible en <https://www.youtube.com/watch?v=chR6kx4btDQ>
- [72] OBS Business School, << Costos directos e indirectos de un proyecto>> [En línea] Disponible en <https://www.obs-edu.com/es/blog-project-management/viabilidad-de-un-proyecto/costos-directos-e-indirectos-de-un-proyecto>
- [73] Wageindicator, sitio web [En línea] Disponible en <https://wageindicator.org/>
- [74] Google Play, << Contenido restringido. Aplicaciones de juegos de apuestas>> [En línea] Disponible en https://play.google.com/intl/es_ALL/about/restricted-content/gambling/gambling-apps/
- [75] Google Play, << Contenido restringido. Sustancias no aprobadas>> [En línea] Disponible en https://play.google.com/intl/es_ALL/about/restricted-content/unapproved-substances/
- [76] Google Play, << Obtención de ingresos y publicidad>> [En línea] Disponible en https://play.google.com/intl/es_ALL/about/monetization-ads/
- [77] Ayuda de Políticas de Google Ads, << Juegos de apuestas>> [En línea] Disponible en <https://support.google.com/adspolicy/answer/6018017?hl=es>
- [78] Boletín Oficial del Estado, Ley Orgánica 3/2018 de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales (BOE núm. 294, de 6 de diciembre de 2018) [En línea] Disponible en <https://www.boe.es/buscar/doc.php?id=BOE-A-2018-16673>
- [79] Boletín Oficial del Estado, Directiva del Consejo 95/46/CE, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos (DOUE L



- núm. 119, de 4 de mayo de 2016) [En línea] Disponible en <https://www.boe.es/buscar/doc.php?id=DOUE-L-2016-80807>
- [80] Real Decreto 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia (BOE núm. 97, de 22 de abril de 1996) [En línea] Disponible en https://www.boe.es/diario_boe/txt.php?id=BOE-A-1996-8930
- [81] Ley 9/2014 de 9 de mayo, General de Telecomunicaciones (BOE núm. 114 de 10 de mayo de 2014) [En línea] Disponible en <https://www.boe.es/buscar/act.php?id=BOE-A-2014-4950>
- [82] Wikipedia, « Diversidad funcional» [En línea] Disponible en https://es.wikipedia.org/wiki/Diversidad_funcional
- [83] FAMMA Cocemfe Madrid, « Diversidad funcional» [En línea] Disponible en <http://famma.org/diccionario-tecnico/418-diversidad-funcional>
- [84] M. Baños-García, F. Lezcano-Barbero, E. García-Maté, R. Casado-Muñoz. “Aplicaciones informáticas dirigidas a jóvenes y adultos con discapacidad intelectual para el desarrollo de las áreas de apoyo”. *Siglo Cero*. 23 Feb 2018. 48(2): 7-24. [En línea] Disponible en: <http://revistas.usal.es/index.php/0210-1696/article/view/14920>
- [85] Android Developers, « Centro de políticas de desarrolladores» [En línea] Disponible en https://play.google.com/intl/es/about/developer-content-policy/#!?modal_active=None



ANEXO A. Carta de la Fundación Prodis



Impresiones sobre DiverApp

El pasado 21 de mayo los alumnos del Programa Avanzas, de la Fundación Prodis, y sus profesores tuvimos la oportunidad de probar la aplicación que Daniel está desarrollando para la estimulación cognitiva de personas adultas con discapacidad intelectual.

Tal y como Daniel pudo comprobar los alumnos se manejaron fácilmente con la aplicación, y disfrutaron de las actividades y juegos que se ofrecían en ella.

Desde el punto de vista de los profesores que les acompañamos, nos gustó ver una aplicación de este tipo con un interfaz adulto y accesible. Es habitual que las aplicaciones que trabajan estos contenidos, similares a los de educación primaria, suelen tener un carácter infantil, y por tanto poco atractivo para nuestros jóvenes de entre 19 y 23 años.

Por otro lado, saber que se trata de una aplicación de la que podríamos obtener datos y estadísticas para ver la evolución de los alumnos, nos gusta de cara a integrarla como apoyo a programas formativos como el de Avanzas.

Como propuestas, creemos que el contraste de los colores y las letras de la aplicación podría mejorarse, ya que la mayoría de los problemas visuales de nuestros alumnos se solucionan en gran medida si los contrastes son buenos.

Por otro lado, a medida que la aplicación evolucione incluir más niveles en cada juego ayudará a que los alumnos puedan seguir mejorando y aprendiendo con ella. Así como, también sería interesante iniciar cada juego con un ejemplo en el que se explique qué tienen que hacer en la actividad, o ventanas emergentes con texto y voz que repitan las instrucciones, un botón de ayuda siempre visible en cada pantalla...

Respecto a las temáticas siempre nos parecen interesantes incluir temas relacionados con las emociones y las habilidades sociales, que pudieran incluso incorporar juegos por equipos.

En general, se trata de una aplicación novedosa que incorpora contenidos muy adaptados al nivel de nuestros alumnos, con algo que la diferencia de las demás, su carácter adulto. ¡Estando encantados de implementarla en un futuro!

PROGRAMA AVANZAS



ANEXO B. Glosario

- ADT (Android Development Tools): plugin utilizado para llevar a cabo el desarrollo de aplicaciones Android en Eclipse.
- Android Studio: entorno de desarrollo integrado (IDE) utilizado para el desarrollo de aplicaciones para el sistema operativo Android.
- Android: sistema operativo desarrollado por Google con el objetivo de poder implantarlo en los dispositivos móviles con pantalla táctil.
- API: herramientas que los desarrolladores tienen disponibles para implementar cualquier funcionalidad de Android en una aplicación.
- APK: archivo que contiene un ejecutable de la aplicación.
- ART (Android Runtime): máquina virtual que realiza la compilación en tiempo de instalación.
- Base de datos: lugar donde se almacena información de un mismo contexto para que posteriormente pueda ser utilizada.
- Bytecode: instrucciones que una máquina virtual espera recibir.
- Código abierto: código fuente de carácter público.
- Coeficiente intelectual: indica la capacidad mental para aprender, razonar y resolver problemas que tiene una persona.
- Compilar: transformación de un lenguaje de programación a otro.
- Cromosoma: estructuras formadas por ADN y proteínas que son contenedoras de la mayor parte de la información genética de un ser vivo.
- Cursor: elemento que almacena toda la información de una consulta a una base de datos.
- Dalvik: variación del lenguaje java que se basa en la compilación en tiempo de ejecución JIT (Just in Time).
- Debug: proceso mediante el cual se pueden visualizar y corregir errores de forma más sencilla.
- Deficiencia neuromuscular: trastornos relacionados con el sistema nervioso de la persona en cuestión.
- Deficiencia osteoarticular: alteraciones motoras y motrices en diversas partes del cuerpo de la persona.
- Dependencias: serie de paquetes que se han de implementar para que la aplicación funcione correctamente.



- Diagrama EDT: es el método mediante el cual se pueden mostrar de forma jerárquica cada una de las actividades o tareas que componen un proyecto.
- Directorio: contenedor virtual donde se almacenan una serie de archivos de carácter informático.
- Discapacidad intelectual: según la OMS, son las personas que puedan tener dificultades para comprender, aprender y recordar cosas nuevas, que se manifiestan durante el desarrollo, y que contribuyen al nivel de inteligencia general, por ejemplo, habilidades cognitivas, motoras, sociales y de lenguaje.
- Drag and drop: proceso mediante el cual el usuario selecciona un objeto de la pantalla y puede arrastrarlo hasta un lugar y soltarlo.
- Drivers: programa informático de un dispositivo encargado de interactuar con un equipo externo.
- Eclipse: plataforma de software de código abierto que cuenta con las herramientas precisas para llevar a cabo el desarrollo de proyectos.
- Enzima: son las encargadas de que las reacciones químicas de un ser vivo sean lo más aceleradas posibles.
- Firebase: plataforma de desarrollo móvil y desarrollo web que ofrece servicios de gran potencial.
- Garbage Collector: sistema de gestión de memoria en el cual se reserva el espacio de memoria necesario y tras haber sido utilizados, se liberan para llevar a cabo una compactación de los espacios de memoria que aún se utilizan y así, dejar todos los libres juntos a la espera de volver a ser utilizados.
- GPS: sistema que permite la localización de cualquier lugar, objeto o persona en la Tierra.
- Gradle: plugin de un sistema de compilación basado en JVM.
- GUI (Graphical User Interface): interfaces gráficas de usuario.
- Hormona tiroidea: regula el metabolismo de la persona.
- IDE: entorno de desarrollo integrado, es decir, la aplicación informática que proporciona al desarrollador las herramientas necesarias para llevar a cabo un proyecto.
- IMSERSO: Instituto de Mayores y Servicios Sociales.
- IU: interfaz de usuario. Medio a partir del cual el usuario puede comunicarse con el sistema.
- JDK (Java Development Kit): es el kit de desarrollo java que hay que instalar para poder hacer uso de determinadas herramientas durante proceso de programación de una aplicación.



- JIT: traduce el bytecode a lenguaje de máquina en tiempo de ejecución.
- JVM: máquina virtual capaz de ejecutar código generado por el compilador de lenguaje Java.
- Keystore: clave criptográfica que dificulta la extracción de una aplicación de un dispositivo.
- Linux: sistema operativo de código abierto y software libre.
- Logcat: herramienta de Android Studio que una serie de mensajes del sistema.
- MeaningCloud: empresa que ofrece los servicios software necesarios para realizar un análisis semántico de un texto.
- Modelo multidimensional
- Open Handset Alliance (OHA): alianza de 84 empresas que se dedican a generar código abierto para dispositivos móviles.
- Play Store: plataforma de distribución de aplicaciones móviles para el sistema operativo Android.
- SDK: lugar donde se almacenan todas las herramientas que se utilizan para llevar a cabo el desarrollo de una aplicación informática.
- Software libre: el código puede ser utilizado para cualquier fin.
- SQLite: biblioteca en lenguaje C utilizado como motor de base de datos SQL caracterizada por ocupar poco espacio, no depender de factores externos, ser rápida y ser fiable.
- Teleeducación: inclusión de las nuevas tecnologías en el proceso de enseñanza-aprendizaje.
- Toast: pequeños carteles que son visibles en la pantalla.
- Valor de tipo primitivo: representa el valor de un tipo de dato char, String, int, boolean.



ANEXO C. Summary

Nowadays exist some foundations whose principal objective is the special education of people with intellectual disability and what is more, they use the new technologies for it.

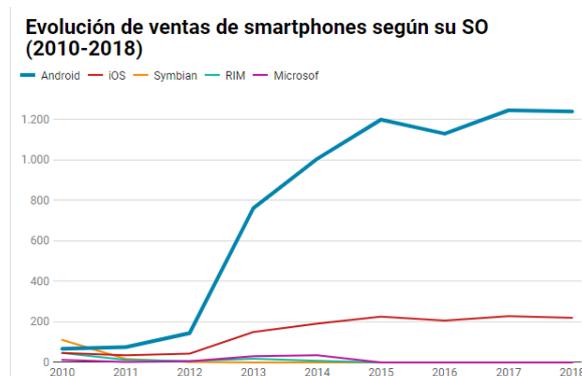
So, in this final project has been developed an Android mobile application that will help to people that suffering slight intellectual disability to do activities with some degree of autonomy. Through this application, users will practise a lot of aspects of the linguistic, mathematical, cultural and self-management area. What is more, has been tried to be a very simple and intuitive application in such way that users could use it when there is no person in charge of them.

C.1 Principal elements of the application

Next is described the principal elements of the developed application besides a a short explanation of the intellectual disabilities.

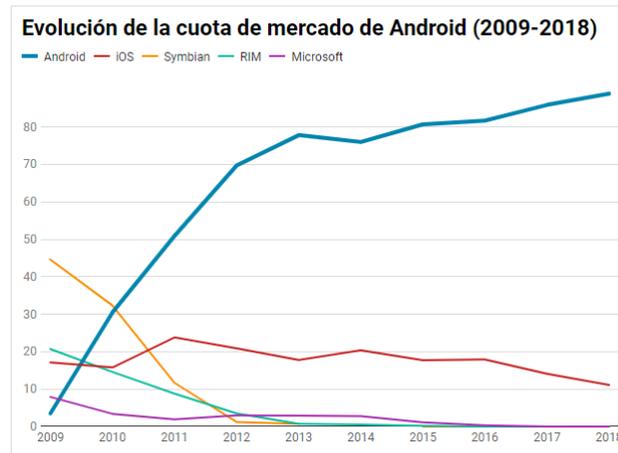
1.1 Android:

The application has been developed for Android devices because after a very huge rise, at present is the number one operative system in sales of smartphones. You can see it in the illustration C.1.



C. 1. Smartphones sales evolution

Besides, Android has experimented a great rise in the operative system market. In 2009 Symbian was the leader but after the appearance of the first SDK Android, all the operative systems except Android have decrease their value market. As you can imagen, iOS is the only one that compete against Android. You can see it in the illustration B.2.



C. 2. Value market evolution

There are a lot of versions of Android and all of them, as curiosity have the name of a sweet. Each one Android version have new functionality and they fix some errors regarding to previous versions. Nevertheless, as sprouts new versions, there are more and more devices that don't support them because their hardware were designed to work whit previous versions. For this application has been selected the version 8.1 which corresponds with API 27 because at this moment is the most used Android version.

Android architecture is composed of four layers. Linux Kernel is the first of them and allows security functions and behaves like an abstraction layer between the hardware and the rest of the software stack. Above Linux Kernel, there is the virtual machine Android Runtime, where the compilation makes in installation time and not in execution time. The native C/C++ libraries that offers some services are in the same layer of the Android Runtime (ART) virtual machine. In the next level of the stack is the framework, where are located all the principal tools for develop an application. These are the APIs, Application Programming Interface, where all programmers have access. At the top of the stack there is the layer where are located the applications of the system that is the basic applications of the device that you can use from another downloaded application, like the camera.

In Android programming there are two very important files. One of them is the APK or Android Package which is the result of compiling (Gradle) and wrapping an application. Inside this package you can find an executable. On the other hand, we have the manifest, which always must be one in an application because takes some information for executing the code, like permissions, name of the application, icon, components, classes, etc.

1.2 Android Studio

The Integrated Development Environment (IDE) used for this project has been Android Studio which have a lot of advantages comparing whit Eclipse ADT, which

is consider an obsolete product because there are a lot of tools that you cannot use in this last IDE Them are:

- Have tools that allows to the developer to write the code more quickly.
- Gradle compiling allows to generate an APK which a good adaptation to the device.
- With Android Studio, the code generated will be the best possible code to programme.
- You can use code in a language that not necessarily have to be java.
- Thera are a lot of Graphical User Interface (GUI) tools.

In the structure of an Android Studio project you can find three directories. The first of them is called 'java' and is where the developer save all *.java* files. In second place, there is 'res' directory, whare you can save all the images, fonts, animations, layouts and menus that you use in your application. Finally, manifests directory is where you can find the Android manifest file described before.

Also there are all Gradle scripts from where you can configurate your own compilation.

Also there is a place where all the tools that you use for the development of an applicaciona are saved. It is the Software Development Kit (SDK) and you can download and update all the versions and that you want from Android SDK Manager.

Finally, Android Studio have a very usefull tool inside. There is a virtual emulator that developers use for making simulations about his code implementation in Android devices in which you can change their type, size of the screen and resolution. You can configurate you virtual device from Virtual Device Configuration menu of Android Studio.

1.3 SQLite

It is a language C library that is used like SQL database engine using very few spaces, been fast, not depending of external factors, been complete and reliable. It has been integrated in mobile phone applications that we use daily.

SQLite databases have been used in the develop of the application because it has some characteristics that differ of other databases managers:

- It is of open source. That is that the user can access to the code with absolutely freedom.
- It is not necessary to configurate previously.
- Doesn't need server support.



- Generate a file that is a diagram of the database. With it, security of the application rises because any other external service can access to inner information.

The application developed is formed with some games and each of them must load questions, possible answers, a code to identify the correct answer and, in some of the games, references to images that are saved in the drawable subdirectory. All of them have been inserted into their respective tables of an SQLite database and, when a user is going to play, the system makes a request to a specific table for get all these elements.

1.4 MeaningCloud

It is a company that offers software services necessary to make semantic analysis of a text. In this process the service that provides MeaningCloud can do some of the next tasks:

- Grammatical labelled: each one of the words that compound a text are classified in a grammatical category.
- Clustering: it makes an analysis of some texts and get the differences between them. Then, classifies the texts. It is used for locale duplicated texts in a document.
- Classification: the analysed text is assigned to one or more categories. You can know the main matter of the text.
- Information extraction: MeaningCloud detect names of people, companies, abstract concept, places, etc. These words provide more information.
- Sentiment analysis: detect the text polarity. If it is very positive, only positive, neutral, negative, very negative or none polarity.

So, in the application developed has been used the service of sentiment analysis of a text that provide MeaningCloud. There is a moment when the character or mascot of the application ask in a dialogue box to the user what is his opinion about a game. For closing the dialogue, the system sends the text answer to the sentiment analysis service of MeaningCloud and gets its polarity. According to the result of the analysis and knowing if the reply of the user is positive or negative, the character of the application will finish the conversation with the suitable response.

1.5 Firebase

It is a mobile developing platform that provides to their clients some great potential services that classifies in three groups.

In the first of them you can find all the tools that allows to compile better and have more security in your applications. These are Coud Firestore, Clode Functions, Authentication, Hosting, Cloud Storage and Realtime Database.



Furthermore, Firebase offer to their users some tools that take statistic and efficiency information for using more efficiently the resources. These tools are Crashlytics, test lab and efficiency supervision.

Finally, in the first group you can find all those tools or services that generate a business rise like In-App Messaging, Google Analytics, Predictions, Cloud Messaging, Remote Config and Dynamic Links.

This platform can be used in some programming languages and not all of them have the same services. Android and iOS are the operative systems that more services have.

Also, Android Studio provides you an assistant for connecting your Android project with your Firebase project. The only thing that you must to do for accessing to the Firebase Assistant is going to the path *Tools>Firebase* through the toolbar.

In the application has been used two services from Firebase:

- **Authentication:** when you initiate the application, appears a screen where you have to fill two gaps, one for your e-mail and the other for your password. Whenever you want to access to the application you must introduce the e-mail and the password that you inserted when you did your signing. In this way, the system differentiates between all the users.
- **Realtime Database:** with the previously service described the application knows who entered in the system. Now, the next step is storing all the opinions and points of the users in an independent way. It means that points and opinions of the games of User A saves in a different place that the points and opinions of User B. It is because has been used Realtime Database, for distinguishing between one user and other.

1.6 Text to Speech Synthesizer (TTS)

It is a process where a digital text is reproduced by a device. For example, a lot of GPS applications used it. While you are driving, how you cannot look to your device screen, the system reproduces an inner digital text informing about the next indications for arriving to your destination. Also, Google Translator used it when you want to hear the translation that you have just do.

The class `TextToSpeech` from Android provides to the developers several parameters that you can configure for introducing in your application a text to speech synthesizer. The only thing that you have to do is to import '*android.speech.tts.TextToSpeech*' package.

Some people with intellectual disability have eyesight problems and sometimes they can't read word or sentences if their background dye is very saturated. So, besides be warned about it, it is because has been decided to introduce this element.



1.7 Intellectual disability

According to World Organisation Health, the intellectual disability is all person that have problems for learning, remembering and understanding new things, that manifest over the growth process, and contribute to the general intellectual level.

Diagnostics and Statistical Manual of Mental Disorders V (DSM-V) classifies the mental disorders providing descriptions for after been diagnosed and learned. The ability for living in society, moving to places, dominating some abilities and learning an employ will depend about which disability is about. DSM-5 classifies them in slight, moderate, serious and deep intellectual disability.

Factors that influence in mental disorders are organic factors (prenatal, perinatal and postnatal factors) and Ambiental factors. Also, there are some other disorders that lead to intellectual disability like Down syndrome, fragile X syndrome, congenital hypothyroidism, Prader- Willi syndrome, etc.

Finally, according to the intellectual disability level, the person will receive more or less help. There are four levels of assistance: intermittent, limited, extensive or generalized.

C.2 Application

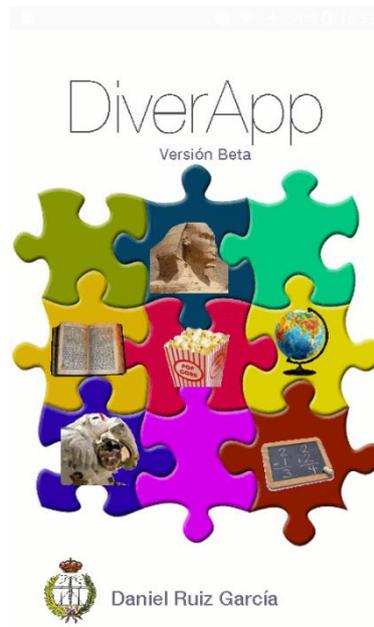
It calls **DiverApp** and it comes from the concept ‘**Diversidad funcional**’ in Spanish which means that all people have different capacities and they must be managed in a way that doesn’t produce exclusion or social discrimination.

DiverApp has five principal areas of practice: linguistic, mathematical, cultural, English and self-management areas.

In linguistic area the user will be able to practice sentences compositions, text compression and memory. In Mathematical area you can practise adds, remainders, fractions and euro management. Also, in cultural area the user will answer questions about sports, geography, art, history, literature and science. How only there is one version of the application and is in Spanish, there is an English area for improve it. Finally, in self-management section, user will be able to practice with clocks, planification of tasks and the resolution of problems.

In illustration C.3 you can see the initial screen of the application.





C. 3. Application

C.3 Evaluation of the App

As part of the project, at 21st May 2019, the developer went to a foundation for confirm that the contents of the application adjust to the specific area of slight intellectual disability.

The visit besides been a great experience, was a success. Professionals of the special education told to me that it was a very good application and that it is the perfect difficult level for them. Also, I could see how all students were very excited with the games. When the session ended, the students filled a survey and the result was excellent. In illustration C.4 you can see one image of the presentation of the application in the foundation.



C. 4. Presentation of the application

C.4 Budget

Have been estimated the costs of all the materials and salaries for calculate how much would have cost this project and the result has been SEVEN THOUSEND SEVEN HOUNDRED SEVENTY-EIGHT WITH THRITY-TWO EUROS.

C.5 Conclusion

The main part of the objectives detailed in this document have been carry out. Besides, as I said before, the result of the surveys has been successful. So, it can conclude that the balance of this Final Degree Project is very positive.