Proyecto Final de Grado

Desarrollo de una aplicación para dispositivos móviles para el aprendizaje y la memorización

Development of an application for mobile devices for learning and memorization

Titulación: Desarrollo de Aplicaciones Multiplataforma

Autor: Eduardo Martínez Ramírez

Tutor: Damián Sualdea Soy.

Índice de contenidos.

Abstract

When we face for the first time a subject of study (a subject, another language, a computer language, driving test), we experience difficulties in memorization due to the large amount of information that must be retained to pass a test of knowledge and skills, and we would like to have some help in memorizing the new vocabulary.

This work (TFG) aims to serve as an aid to the memorization of concepts and technical terms used in the development of mobile applications on the Android platform, and to try to further emphasize the learning of this language, it is proposed to develop this application using the Android system itself, This has been a great challenge for the student (the author of the application), since it has been difficult to separate his point of view as a developer (having already passed the formative subjects, and in particular that of Multimedia Programming and Mobile Devices), with his point of view as a user of the application, as they almost overlap temporarily.

Although the initial ambition was to create an app of almost commercial quality, very versatile (not only for learning Android but also for other subjects), very configurable by the user (not only for mobiles but also for tablets, and even for desktop computers), we soon had to recognize the limitation of resources (especially the time available) and we had to stick to a specific case.

The specific objective of this work is to design and develop an Android app to review and memorize concepts of the Android operating system.

Resumen y objetivos

Resumen:

Cuando nos enfrentamos por primera vez a una materia de estudio (una asignatura, otro idioma, un lenguaje informático, el examen de conducir), se experimentan dificultades en la memorización por la gran cantidad de información que debe retenerse para superar un examen de conocimientos y habilidades, y se desearía disponer de alguna ayuda a la memorización del nuevo vocabulario.

Este trabajo (TFG) pretende servir de ayuda a la memorización de conceptos y términos técnicos usados en el desarrollo de aplicaciones móviles en la plataforma Android, y para tratar de recalcar aún más el aprendizaje de este lenguaje, se propone desarrollar esta aplicación utilizando el propio sistema Android, lo que ha supuesto un gran reto para el alumno (el autor de la aplicación), ya que le ha sido difícil separar su punto de vista de desarrollador (ya superadas las asignaturas formativas, y en particular la de Programación multimedia y dispositivos móviles), con su punto de vista como usuario de la aplicación, pues casi se superponen temporalmente.

Aunque en un principio se ambicionaba crear una app de calidad casi comercial, muy versátil (que no solo sirviese para aprender Android sino otras asignaturas), que fuera muy configurable por el usuario (que sirviese no solo para móviles sino para tabletas, e incluso en equipos de sobremesa), enseguida hubo que reconocer la limitación de recursos (sobre todo de tiempo disponible) y hubo que ceñirse a un caso concreto.

El objetivo concreto de este trabajo es diseñar y desarrollar una app en Android para repasar y memorizar conceptos del sistema operativo Android.

Motivaciones

Conforme se ha avanzado y estudiando las asignaturas en este módulo, han surgido cada vez más dificultades a la hora de memorizar y aprender muchos textos técnicos, los cuales exigen mucho rigor y un entendimiento claro de su sintaxis como en el caso del código en Java en la asignatura de programación, o en la asignatura de programación multimedia y dispositivos móviles en la cual se aprende Android.

Debido al conocimiento adquirido en esta última asignatura, y a la dificultad en su nomenclatura, se decidió desarrollar una aplicación que pudiera ayudar en tareas de aprendizaje y, sobre todo, de memorización.

Objetivos

El objetivo de esta aplicación es brindar ayuda a futuros estudiantes, permitiéndoles estudiar en cualquier lugar y momento si disponen de un dispositivo móvil compatible con Android.

La aplicación contará con una interfaz sencilla e intuitiva con la cual podrán guardar preguntas y respuestas. Gracias a ello podrán poner a prueba sus conocimientos de una manera rápida y organizada, sin necesidad de crear formularios, como suele hacerse en un sin fin de herramientas las cuales intentan aportar estas soluciones.

Antecedentes

La memorización de términos técnicos (y de vocabulario en otro idioma) es una tarea complicada. Para ayudarnos, se puede:

- Dividir el material en partes más pequeñas (ej. por idiomas o lenguajes de programación, por temas...).
- Escribir (anotar) la información que se desea recordar (una vez que se ha estudiado el tema).
- Crea ayudas visuales, como diagramas o gráficos, que te ayuden a comprender y recordar.
- Repasar regularmente para asegurarse de que estás reteniendo la información, pero con descansos y cambios de tema, para prevenir la saturación y fatiga mental.
- Crear recursos mnemotécnicos, como acrónimos, rimas, canciones, o juegos, que te ayuden a recordar.

Las apps (Apps, 2023) pueden ayudar con la memorización de términos técnicos de varias maneras. Una forma es mediante el uso de una aplicación móvil de tarjeta didácticas (Tarjeta didácticas, 2023) o *flashcards* (Flashcard, 2023) a las que se puede acceder en cualquier momento y lugar, y permiten a los usuarios revisar rápida y convenientemente términos técnicos, vocabulario, definiciones...

Las flashcards son tarjetas que contienen palabras, imágenes, símbolos o números en uno o ambos lados y se usan para adquirir conocimientos al memorizar su contenido mediante el repaso espaciado del conjunto de tarjetas. Una de las variantes originales de este método se conoce como Sistema de Leitner (Sistema de Leitner, 2023).

El método clásico de uso de la técnica de tarjetas didácticas consistía en preparar trocitos de cartulina (el papel no aguantaba muchos manejos) del mismo tamaño, donde se escribía por un lado la pregunta, y por el otro la respuesta. Una variante era doblar por la mitad las cartulinas y

escribir pregunta y respuesta en el mismo lado, lo que permitía iniciar el aprendizaje viendo a la vez cada pregunta con su respuesta, para luego doblar la cartulina y recuperar la estructura de ver solo la pregunta (o solo la respuesta).

Las tarjetas didácticas pueden contener imágenes (y en las apps, otras ayudas multimedia, como audio o vídeo), ej. para aprender contornos de mapas u otras figuras geométricas, aprender banderas y otras señalizaciones gráficas, etc. Particularmente para el aprendizaje infantil, las imágenes son fundamentales, y las apps permiten incluso animación (no solo vídeo, sino animación de esquemas educativos y realidad virtual).

Una app de aprendizaje puede proporcionar pruebas y cuestionarios interactivos que pueden ayudar a los usuarios a evaluar su comprensión del material: idiomas, biología, anatomía, geografía, historia, literatura, matemáticas... También pueden proporcionar ayudas de audio y visuales que ayuden a los usuarios a comprender y recordar mejor los términos técnicos.

Existen otras herramientas que sirvieron de inspiración a la hora de desarrollar esta aplicación; una de las más estudiadas fue la de los formularios de Google docs (Google docs, 2023). Con estos formularios, es fácil diseñar preguntas tipo test con varias opciones y compartir el formulario a través de un enlace. Destaca por tener una interfaz limpia, de fácil manejo y muy configurable, además de poder ajustar múltiples valores como, por ejemplo:

- Cambiar el orden de las preguntas.
- Configurar el límite de intentos en el formulario.
- Ajustar puntuaciones.
- Recoger estadísticas de los participantes.
- Gestionar colaboradores y dar privilegios para cambiar el formulario.

A pesar de sus ventajas, esta herramienta no nos pareció perfecta y presentaba diversos problemas.

En escalabilidad, por ejemplo, si se desea realizar un test con 70 preguntas, sería engorroso tener que responder a todas esas preguntas para probar los conocimientos, o dejarlas en blanco, lo cual implicaría una pérdida de tiempo considerable si no se puede supervisar cada una de las preguntas individualmente. Otra solución sería dividir esas 70 preguntas en 7 formularios diferentes de 10 preguntas, pero provocaría que no hubiese variedad y que fuese muy fácil de saber las preguntas simplemente por repetición sin prestar atención al enunciado.

Otro problema grave es que únicamente funcionan para exámenes tipo test, sin poder desarrollar explicaciones ni prepararse para diferentes tipos de examen.

También se probaron otras herramientas durante el curso, como el ‘Kahoot!’ y el ‘Quizizz’ ambas similares que destacan en su interfaz fluida y en su sistema de puntuación llamativo con diferentes *powerups* (poderes con los que aumentar tu puntuación según el tipo de reto).

A pesar de sus muchas virtudes, algunas personas pueden considerar que estas herramientas son un tanto infantiles cuando se utilizan para estudiar temas avanzados, ya que cuentan con muchos colores y animaciones que pueden alejarse de su finalidad principal.

También se probaron otras aplicaciones disponibles en la Play Store que pueden ayudar en la memorización y el aprendizaje de diferentes temas como:

- (AnkiDroid, 2023). Esta aplicación utiliza el método de tarjetas de memoria para ayudar en la memorización de conceptos. Permite crear y organizar tarjetas con preguntas y respuestas, y utiliza técnicas de repetición espaciada para optimizar el proceso de aprendizaje.

- (Quizlet, 2023). Es una plataforma muy popular que ofrece una amplia variedad de herramientas de estudio, como tarjetas de memoria, juegos interactivos, exámenes y pruebas de práctica. También cuenta con una gran base de datos de contenido compartido por otros usuarios.
- (Memrise, 2023). Es una aplicación que se centra en la memorización de vocabulario y conceptos mediante la repetición y la asociación visual. Utiliza técnicas de gamificación para hacer el aprendizaje más entretenido.
- (Duolingo, 2023). Aunque inicialmente se desarrolló como una aplicación para aprender idiomas, Duolingo también puede ser útil para la memorización de vocabulario y conceptos. Ofrece lecciones interactivas, ejercicios de traducción y repeticiones para reforzar el aprendizaje.

Hemos probado a usar algunas de las apps de flashcards disponibles, y todas se parecen un poco, basándose en un conjunto de plantillas entre las que el usuario elige una, y dentro de la cual puede cambiar alguna configuración sencilla, como tipo de letra o color de fondo.

Habiendo tantas apps de desarrollo de flashcards, ¿por qué esforzarse en desarrollar una más?

- Porque esto es un PFG que debe ayudar a la formación del alumno autor de la app, y no un desarrollo con fines comerciales.
- Porque casi todas las apps generadoras de flashcards se basan en plantillas cerradas, y aquí queremos tener más libertad de diseño (para profundizar en el conocimiento de aplicaciones móviles).
- Aquí se pretende desarrollar una app específica, original, muy personalizada, y por tanto difícilmente configurable por el usuario una vez que el programador ha fijado el diseño. Sería una app diseñada a medida de un usuario, no una app genérica de flashcards.

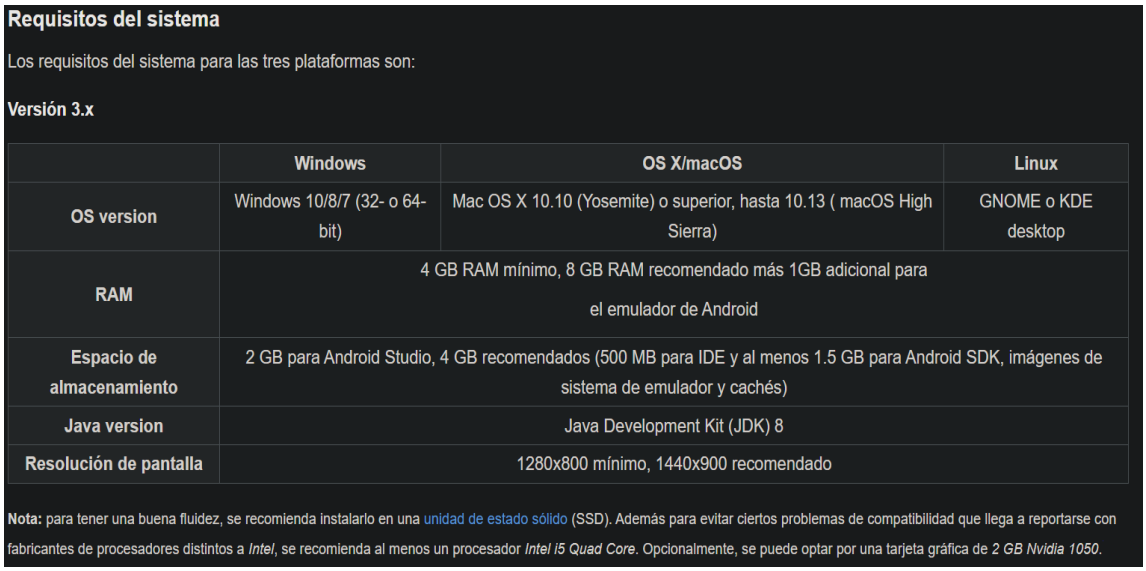
Análisis y especificación de requisitos

Se trata de una app en Android para repasar términos técnicos del propio lenguaje Android. Será una app nativa (que corre en el móvil), autónoma (totalmente contenida en el móvil, sin necesidad de conexión externa), que se programará en un equipo de sobremesa usando como entorno de desarrollo integrado Android Studio (versión 'Electric Eel' de 2022, o superior), con un objetivo targetApi=31 o superior para cumplir con los requisitos de Google Play.
Nótese que, según Google Play (Google Play, 2023), a partir de agosto de 2023, las apps nuevas se deben orientar a Android 13 (nivel de API 33) o versiones posteriores. Actualmente, las apps existentes (para dispositivos móviles, Android Auto y Android TV) deben orientarse al nivel de API objetivo 31 o versiones posteriores antes del 31 de agosto de 2023 (nivel de API objetivo 30 para Wear OS). De lo contrario, dejarán de estar visibles para todos los usuarios de Google Play cuyos dispositivos ejecuten versiones del SO Android más recientes que el nivel de API objetivo de tu app, ya que esta no se diseñó para cumplir con los estándares de seguridad y calidad que esos usuarios esperan de las versiones más recientes del SO Android.

Como dispositivo mínimo requerido para correr la app, se establecerá como Minimum SDK 'API 29', es decir los dispositivos con SO Android versión 10 o superior, pese a que Android Studio indica que con ello esta app solo se podrá ejecutar en el 68 % del parque de móviles existentes actualmente.

Como en la mayoría de las aplicaciones de flashcards, el contenido de las tarjetas de estudio se creará también en una versión de escritorio, directamente en Android Studio, o con ayuda de un editor de CSV, o en Excel; no en la propia app móvil, debido a la complejidad de las preguntas que se piensa formular.

Para poder realizar la app necesitaremos instalar Android Studio en nuestro ordenador por lo que deberemos tener los siguientes requisitos:



Requisitos del sistema

Los requisitos del sistema para las tres plataformas son:

Versión 3.x

	Windows	OS X/macOS	Linux
OS version	Windows 10/8/7 (32- o 64-bit)	Mac OS X 10.10 (Yosemite) o superior, hasta 10.13 (macOS High Sierra)	GNOME o KDE desktop
RAM	4 GB RAM mínimo, 8 GB RAM recomendado más 1GB adicional para el emulador de Android		
Espacio de almacenamiento	2 GB para Android Studio, 4 GB recomendados (500 MB para IDE y al menos 1.5 GB para Android SDK, imágenes de sistema de emulador y cachés)		
Java version	Java Development Kit (JDK) 8		
Resolución de pantalla	1280x800 mínimo, 1440x900 recomendado		

Nota: para tener una buena fluidez, se recomienda instalarlo en una unidad de estado sólido (SSD). Además para evitar ciertos problemas de compatibilidad que llega a reportarse con fabricantes de procesadores distintos a *Intel*, se recomienda al menos un procesador *Intel i5 Quad Core*. Opcionalmente, se puede optar por una tarjeta gráfica de *2 GB Nvidia 1050*.

También deberemos tener conocimientos teóricos en Android para poder realizar la app de la manera más eficiente y sencilla posible. He aquí una breve explicación sobre qué es Android:

1. Android (Android, 2023) es un sistema operativo móvil basado en el núcleo Linux (Núcleo Linux, 2023) y otros software de código abierto.
2. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes Wear OS, automóviles con el sistema Android Automotive o con otros sistemas a través de Android Auto, y televisores con Android TV.
3. Inicialmente fue desarrollado por Android Inc., que fue adquirido por Google en 2005. Android fue presentado en 2007 junto con la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles. El código fuente principal de Android se conoce como Android Open Source Project (AOSP), que se licencia principalmente bajo la Licencia Apache. Android es el sistema operativo móvil más utilizado del mundo, con una cuota de mercado superior al 90 % al año 2018, muy por encima de IOS y otros.

Dado que la aplicación de Android Studio admite Java y Kotlin se decidió hacer el desarrollo con Java que es el lenguaje más estudiado durante el curso.

He aquí una breve explicación sobre que es Java.

1. Java (Java, 2023) es un lenguaje de programación y una plataforma informática que fue comercializada por primera vez en 1995 por Sun Microsystems.
2. El lenguaje de programación Java fue desarrollado originalmente por James Gosling, de Sun Microsystems (adquirida en 2010 por la compañía Oracle), y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son compiladas a bytecode (clases Java), que puede ejecutarse en cualquier máquina virtual Java (JVM, 2023) sin importar la arquitectura de la computadora subyacente.
3. La compañía Sun desarrolló la implementación de referencia original para los compiladores de Java, máquinas virtuales y librerías de clases en 1991, y las publicó por primera vez en 1995. A partir de mayo de 2007, en cumplimiento de las especificaciones del Proceso de la Comunidad Java, Sun volvió a licenciar la mayoría de sus tecnologías de Java bajo la Licencia Pública General de GNU. Otros han desarrollado también implementaciones alternativas a estas tecnologías de Sun, tales como el Compilador de Java de GNU y el GNU Classpath.

Propuesta de solución

En concreto, se trata de desarrollar una app en Android de ayuda a la memorización por flashcards del propio sistema Android, particularmente cuando se usa el lenguaje de programación Java.

Se pretende crear una app autónoma (*stand-alone*), para el aprendizaje individual de conceptos técnicos avanzados, para personas adultas (no para educación infantil), y, aunque se podrá poner a disposición de otros usuarios tanto la app completa como la base de datos, no se contempla el uso colaborativo en equipo de estudiantes.

Antes de ponerse a desarrollar la app, hay que prever cómo la utilizará el usuario (el que quiere memorizar). Para empezar, vamos a restringir el modelo de aprendizaje al **test de respuesta correcta única**, y no a test de respuesta múltiple a elegir solo una, o a otros modelos de test.

En concreto la solución que se propone se va a basar en una única pantalla (una Activity) en la que, con un clip del usuario, aparecerá aleatoriamente una pregunta del temario, y el usuario podrá introducir la respuesta que considere oportuna y, mediante un botón de validación, compararla con la respuesta correcta.

Esta app, al cargarse presentará una pantalla de instrucciones de manejo, con un botón para saltar a la app propiamente dicha, que constará de una única pantalla con diversos cuadros de texto y botones con los que el usuario podrá interaccionar con una base de datos de preguntas y respuestas creada por el desarrollador.

- Un campo de texto estará destinado a mostrar la pregunta.
- Un campo de imagen podrá añadirse tras el anterior, si la pregunta va acompañada de información gráfica.
- Un campo de texto editable estará destinado a introducir y mostrar la respuesta.

- Un botón de validación que servirá para comprobar si el usuario ha acertado con su respuesta. Si el usuario no introduce nada, se sobreentenderá que se rinde, y se le mostrará la respuesta.
- Un botón de nueva pregunta introducirá la siguiente pregunta.
- Un botón para seleccionar las preguntas por temas.
- Se intentará enriquecer esta GUI con otros elementos:
    - Un botón de ‘ayuda’ para mostrarle los primeros caracteres de la respuesta (se piensa en 2 o 3 caracteres).
    - Un botón de ‘paso’ (me rindo).
    - Un cuadro de texto donde se muestren conjuntamente la respuesta del usuario y la correcta (la de la base de datos). Esto ayudaría al usuario que a lo mejor se ha equivocado solo en una letra, o la ha puesto en castellano en lugar de en inglés.
    - Un selector del tema de la pregunta, ya que es normal que solo se desee repasar un tema concreto (ej. el primero, si se está empezando el estudio).
    - Un selector del nivel de dificultad de la pregunta.
    - Un botón de ‘quitar’ la pregunta de la memoria (no de la base de datos original), para que no le aparezca más al usuario durante esa sesión, i.e. una especie de ‘esta ya me la sé’.

La app debería proponer las preguntas al azar, pero con orden:

1. Empezando por las más básicas y sencillas, i.e. dificultad incremental.
2. Insistiendo en las que al usuario le han resultado más difíciles de retener (donde más ha fallado).
3. Introduciendo en cada nivel superior, algunas cuestiones de niveles ya superados.

Plan de trabajo

Según las especificaciones, se trata de crear una app en Android para repasar términos técnicos del propio lenguaje Android.

Lo primero de cualquier planificación es marcar plazos y presupuestos. Aquí nos ceñimos a las especificaciones generales del TFG (dos meses y medio de trabajo, en nuestro caso de manera individual, a tiempo compartido con la Formación en Centros de Trabajo), y a las especificaciones de este TFG en particular (presentadas más arriba).

La experiencia en programación durante estos dos años de estudios de FP-DAM, y en la presentación de informes, nos aconseja dejar al menos dos de las diez semanas para el 'pulido' del código y del informe, así que se ha dedicado una semana a recopilar información en Internet sobre aplicaciones similares, otra semana a recopilar datos para luego crear la base de datos a memorizar, y unas cuatro semanas a diseñar la app. Se ha procedido por iteraciones, empezando con una pequeña muestra de preguntas y respuestas, los botones de una interfaz de usuario básica que permitan preguntar y responder, y unas funcionalidades básicas (cambiar de pregunta, permitir responder, y mostrar la respuesta almacenada. Luego se han ido incorporando más preguntas, más temas, y más funcionalidades.

Se prevén dos fases para esta aplicación, que corresponden a funcionalidades muy diferentes por parte del usuario, las cuales podrán desarrollarse en serie (una tras otra), o añadiendo preguntas conforme se va estudiando:

1. Cómo generar la base de datos (BD), es decir las parejas clave/valor de preguntas y respuestas (P/R). Este trabajo aquí lo va a realizar el desarrollador, aunque convendría que lo hiciese el usuario, lo que le ayudará a ir aprendiendo sobre la marcha. En otros casos la base de datos vendría dada por profesores o instituciones (ej. las aplicaciones para memorizar las normas de tráfico para los exámenes de conducir, suelen suministrarlas las autoescuelas, o la propia DGT, y no las desarrolla el usuario, que solo las utiliza).

a. Una solución sería editar las P/R dentro de la app, en un archivo /res/strings/bd.xml de la propia app.

b. Otra solución sería editar las P/R en un archivo CSV o una hoja de cálculo Excel que luego sería leída por la app.

c. Otra solución sería usar una BD SQLite, añadiendo a la app un método de edición aparte.

d. Conviene que, además, de estar identificada cada pregunta con su respuesta, poder agruparlas por temas y por niveles de dificultad.

2. Cómo usar la BD una vez que ya está disponible:

    a. Se creará la opción de un corto tutorial de funcionamiento de la app (de una página o así), además de un manual de uso más completo.

    b. Una primera opción sería que mostrase cada pregunta con su respuesta, para la fase inicial de toma de contacto con la asignatura.

    c. Luego, en la fase principal de ayuda a la memorización, se mostrarán solo las preguntas, y el usuario tendrá que adivinar la respuesta.

        i. El usuario podrá elegir un nivel de dificultad: bajo, medio, alto, o aleatorio.

        ii. El usuario podrá solicitar una ayuda, como que se le muestren las primeras letras de la solución.

        iii. El usuario podrá rendirse y se le mostrará la solución.

        iv. El usuario podrá añadir notas cortas de comentarios suyos a algunas preguntas que, aunque se guardará aparte para preservar la integridad de la base de datos, se mostrarán a continuación de las respuestas.

v. El usuario podrá marcar algunas preguntas como ‘aprendidas’ para evitar que le sigan apareciendo.

vi. Al dar una respuesta el usuario, el programa mostrará la respuesta correcta y la del usuario.

vii. Al cabo de cada sesión, el programa mostrará la relación de aciertos sobre el total, para que el usuario vaya viendo sus progresos. Se podría incluso presentar una estadística del promedio de aciertos a lo largo del tiempo.

No se contempla incluir en esta app otras funcionalidades auxiliares, como envío automático de recordatorios si no has estudiado durante un tiempo, o como establecer competiciones entre diversos usuarios. Tampoco se van a incluir las funcionalidades típicas de selección de tipo y tamaño de letra, colores de texto y de fondo, alineaciones, y demás, dejando estos aspectos cosméticos y de accesibilidad para una etapa posterior del desarrollo.

La primera elección ha sido la del idioma de trabajo, que, pese a que el inglés es el lenguaje franco de la informática, se ha elegido el español.

Se intentará añadir posteriormente algún botón u opción para poder cambiar de idioma (de inglés a español y viceversa), aunque únicamente cambiaría la interfaz, no el contenido de las preguntas ni las respuestas.

A diferencia de las aplicaciones para vocabulario de idiomas, donde tiene sentido poder intercambiar preguntas con respuestas (ej. es tan interesante preguntar cómo se dice en español ‘house’ que cómo se dice ‘casa’ en inglés), aquí no se pueden intercambiar las preguntas (que suelen ser de muchas palabras) con las respuestas (que suelen ser mucho más cortas).

Desarrollo de la solución

Para implementar un modelo de tarjeta flash en Android, se debe crear un ‘Nuevo Proyecto’ en Android Studio, elegir una de las plantillas existentes (se ha usado la plantilla ‘Empty Activity’), elegir nombre para la app (se ha llamado ‘AndroidFlash’), elegir nombre del paquete (se ha dejado el que viene por defecto, elegir ubicación de destino de los archivos, elegir lenguaje (se ha usado Java), y establecer el nivel mínimo de dispositivo compatible con la app (se ha elegido Android 10 (API 29).

El diseño de la interfaz de usuario gráfica (GUI, /res/layout/activity_main.xml) se muestra en la Fig. 1. Las funcionalidades se han implementado en el MainActivity.java. Los listados del código se recogerán en un Anexo. Aquí solo se van a comentar los aspectos destacables.

Uno de los mayores problemas ha sido el continuo rediseño de la GUI para alojar cada vez más botones y campos de texto.

La incorporación de campos de imagen para preguntas que requieren gráficos, no se ha resuelto del todo bien, habiéndonos limitado a media docena de casos por la complejidad del trabajo de acomodación en una sola pantalla.

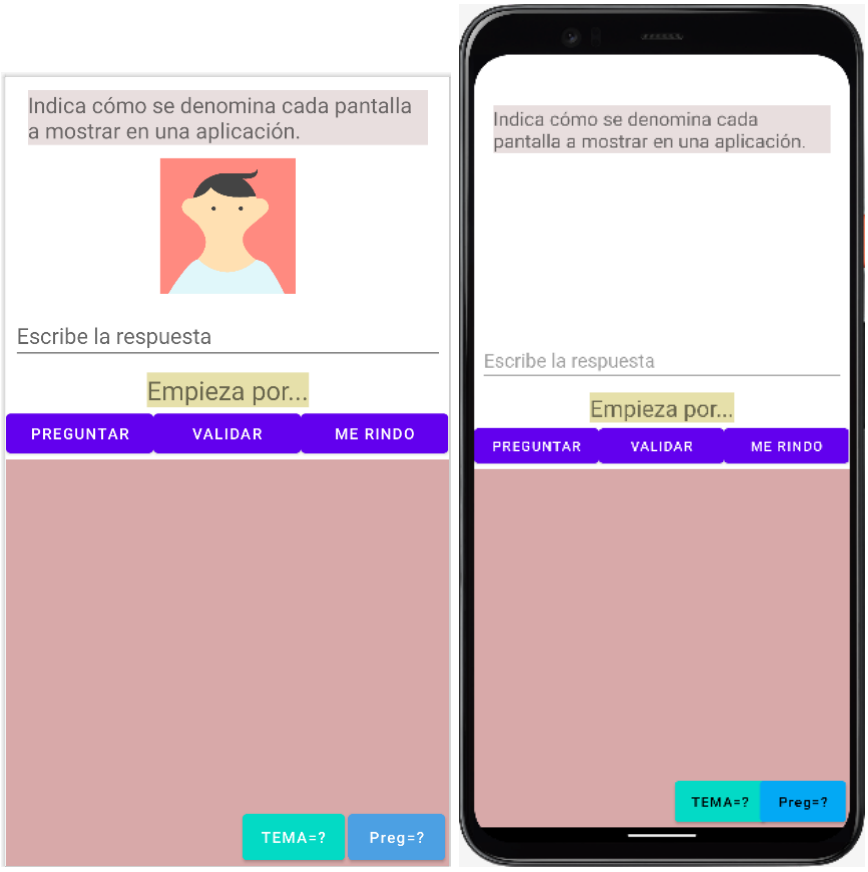


Fig. 1. a) Layout de la GUI. b) Una muestra de la pantalla de la app.

Ahora se va a explicar de manera más detallada el código con el que se ha desarrollado la aplicación:

- La GUI tiene 7 onclicks: b1(preguntar), t2(ayudar), b2(validar), b3(me_rindo), t3(me_rindo), tema(tema), preg(preg). El MAIN tiene, además del onCreate(), estos 7 métodos: almacenar(), preguntar(View v), validar(View v), me_rindo(View v), ayudar(View v), tema(View v), y preg(View v); estos últimos 6 métodos se corresponden con los 7 onclick, ya que dos onclick() apuntan al mismo me_rindo(View v).
- La BD original de Preguntas/Respuestas está dentro de la propia app, en /res/values/strings.xml, junto a otras strings (app_name y cover), como se muestra en el Fig. 2.

```
<resources>
    <string name="app_name">AndroidFlash</string>
    <string name="cover"...>
    <string name="t1_p1">Indica cómo se denomina cada pantalla a mostrar en una aplicación.</string>
    <string name="t1_r1">Activity</string>
    <string name="t1_p2">Componente visual mediante el cual el usuario de alguna manera controla o pro
    <string name="t1_r2">Widget</string>
    <string name="t1_p3">Atributos o propiedades que son necesarias para dar altura y anchura a cualqu
    <string name="t1_r3">layout_width y layout_height</string>
    <string name="t1_p4">Para el layout raíz, el valor de los atributos width y height es:</string>
    <string name="t1_r4">match_parent</string>
    <string name="t1_p5">Para los layouts o widgets internos al layout raíz los atributos width y heig
    <string name="t1_r5">wrap_content</string>
    <string name="t1_p6">Componente no visual destinado a controlar la distribución, la posición y las
    <string name="t1_r6">Layout</string>
    <string name="t1_p7">Componente esencial que utiliza android para moverse de una pantalla (activit
    <string name="t1_r7">Intent</string>
```

Fig. 2. Copia parcial del archivo /res/values/strings.xml.

- Cada pregunta se guarda en formato clave/valor, donde la clave es el nombre de la string (ej. para la 1ª pregunta del 1er tema, la clave es 't1_p1'), y el valor es el contenido de la string. Y de modo análogo para cada respuesta, que se escribe justo debajo (por facilidad, pues el orden no importa).
- Al ejecutar la app, esta BD original se carga con el método almacenar() en el archivo /data/data/com.example.androidflash/shared_prefs/q.xml que es la BD activa que usa la app (la BD original solo se usa en la app para construir la BD de trabajo, con almacenar(); si ya está construida, se puede saltar lo de almacenar()). Este método almacenar() es:

```
public void almacenar() {
  File f= new File("/data/data/com.example.androidflash/shared_prefs/q.xml");
f.delete();
  int i,j,id1,id2; String s1,s2;
  SharedPreferences.Editor spe = pref.edit();
  for (j=1;j<(1+n.length);j++)     //Para cada uno de los temas j
    for (i=1;i<(1+n[j-1]);i++) { //Para cada una de los preguntas de cada tema
      s1="t"+j+"_p"+i; s2="t"+j+"_r"+i;
      id1 = this.getResources().getIdentifier(s1,"string", getPackageName());
      id2 = this.getResources().getIdentifier(s2,"string", getPackageName());
      spe.putString(s1, getString(id1)); spe.putString(s2, getString(id2));}
spe.commit();}
```

  - En la 1ª línea se limpia (borra), si existe, el archivo que contendrá la BD activa, ya que la grabación es por 'añadir al final' (append).
  - En la 2ª línea se declaran las variables locales: los iteradores 'i' y 'j', los identificadores 'id1' e 'id2', y las strings 's1' y 's2' que son las

claves que se generan programáticamente, que coinciden con los nombres de las strings del archivo interno /res/values/strings.xml.

- En la 3ª línea se construye la variable 'spe' de tipo SharedPreferences.Editor que permite grabar en el archivo definido en onCreate() así:

  ```
  pref = getSharedPreferences("q",Context.MODE_PRIVATE);
  ```

  lo cual se lleva a cabo en los dos bucles anidados siguientes del código (el 1º para los temas, y el 2º para las preguntas de ese tema).
- En la última línea se escriben en la BD las parejas P/R, acabando con el 'commit()' que finaliza la grabación.

- Tras almacenar la BD (o si ya está almacenada), la GUI muestra la 1ª pregunta y espera a que el usuario clique en un botón. Las preguntas tienen como clave la variable 'claveP', que puede tomar los valores 't?_p?', siendo '?' el número de orden, y la 't' y la 'p' las iniciales de 'tema' y 'pregunta'. Así que, para seleccionar la 1ª pregunta del 1er tema, y su respuesta (cambiando la 'p' por 'r'), en el MAIN se pone:

  ```
  claveP="t1_p1"; claveR="t1_r1";
  ```

  Nota: si no coincidiera con el texto cargado en la GUI, tras esta selección, se debería cargar esa pregunta en el cuadro de texto de preguntas (t1) con la instrucción:

  ```
  t1.setText(pref.getString(claveP, ""));
  ```
- Tras cargarse la app, el usuario puede elegir entre varios botones, antes o después de introducir su respuesta a la pregunta planteada.
  - Si elige 'preguntar' (b1), se cargará una nueva pregunta (en t1).
  - Si elige 'validar' (b2), se comparará su respuesta (aunque esté vacía) con la solución correcta, y se le indicará (en el t2) si acertó o falló.
  - Si elige 'Empieza por...' se le mostrarán los 3 primeros caracteres de la respuesta (abajo, en el t3). Nota: este botón de ayuda 'Empieza por...', funciona en todo momento, aunque aparezca sobrescrito con el resultado de validar (i.e. 'acertaste' o 'fallaste').
  - Si elige 'me rindo' se le mostrarán la respuesta (abajo, en el t3). Nota: también se le mostrarán la respuesta si clica en el gran bloque de texto t3 (se usó para la depuración, y pareció bien dejarlo así).

- Si clica en el botón ‘tema=?’ irá cambiando circularmente a: preguntas de cualquier tema (la ‘?’), solo del tema 1, del 2, del 3, del 4, o del 5. También se dispone el método siguiente para resetear a una ‘?’ el tema, al realizar una pulsación mantenida en el botón tema:

```
tema.setOnLongClickListener(new View.OnLongClickListener() {
  public boolean onLongClick(View v) {
    t=0;
    b5.setText("Tema=?");
    return true;}});
```

- Si clica en el botón ‘preg=?’ irá cambiando circularmente a: pregunta aleatoria, pregunta 1ª, pregunta 2ª, y así sucesivamente.
  Al igual que en el botón de tema, preg tiene otro método similar para resetear una ‘?’ con una pulsación larga.

Despliegue e instalación

- Se trata de una app en Android (y para Android), así que su ‘mercado’ se limita a dispositivos móviles compatible con Android.
- Esta app funciona en los emuladores de Android Studio y en los dispositivos móviles en los que se haya cargado esta app por uno de estos procedimientos:
    - A través de un cable USB desde Android Studio
    - A través de cualquier medio (cable USB, descarga por Wifi o por Internet) de la APK desde el PC del creador o desde uno de sus almacenes en la nube (ej. GoogleDrive).
- En el futuro, se le pondría un nombre más atractivo y se alojaría en la tienda ‘PlayStore’ de Google, en la sección de software educativo gratuito, para facilitar la búsqueda y poder descarga sin intervención del desarrollador.

  Una vez instalada la app, no se requiere conexión a Internet para su uso, así que puede utilizarse incluso en ‘modo avión’.
- El manual de usuario constará de unas instrucciones básicas, incluidas en la página inicial de la app, conforme se muestra en la Fig. 3.

Fig. 3. Pantalla de inicio de la app, con instrucciones de uso.

Evolución y trabajo futuro

Se ha desarrollado una app en Android de ayuda a la memorización por flashcards del propio lenguaje Android. Se ha priorizado el conseguir una app que funcione, así que somos conscientes de que el primer trabajo tras la entrega de este informe sería redactarlo mejor, pues algunas partes se escribieron al principio, cuando la ambición era grande y nuestros recursos no parecían tener límite, y otras partes se han redactado deprisa y corriendo para cumplir los plazos inexorables.

¿Qué mejoras convendría introducir en esta app en el futuro (aparte de revisar su documentación)? Pues ya las hemos ido apuntando cuando hemos establecido las limitaciones en el plan de trabajo. Por un lado, están las que se refieren a esta app en concreto de memorización de términos técnicos de Android, y por otro lado están las que ampliarían el campo de estudio a otras asignaturas, a otros rangos de edad (e.g. apps de juegos infantiles o de enseñanzas básicas de conocimiento del medio).

En el estado actual de la app, el banco de preguntas es bastante limitado (menos de 100 preguntas) y de carácter más bien básico (apenas hay preguntas de aspectos avanzados como la arquitectura Android, librerías y *frameworks*, diseño de plantillas, etc. Y ni siquiera se ha implementado la clasificación por nivel de dificultad, aunque parece que no sería difícil si se introduce un nuevo control en las claves; ej. si la pregunta 2ª del tema 1º tuviese dificultad 3 (en una escala creciente 1-2-3-4-5), se podría poner ClaveP=”t1_p02_d3” y CalveR=”t1_r02_d3” para seguir con el tratamiento coordinado P/R (nótese que se deben utilizar dos dígitos para el número de la pregunta, para poder tener un acceso sencillo al nivel de dificultad ‘d’, que siempre estaría en la misma posición de las strings.

Una importante mejora sería poder editar y manejar la base de datos desde la propia app por el usuario (sin necesidad de que el programador retoque la aplicación), para ir añadiendo nuevas preguntas; modificar el nivel de dificultad, el tema, o algún enunciado ambiguo (o erróneo); añadir ilustraciones, etc.

También se podría implementar un cuadro de texto para que el usuario introdujese comentarios a cada pregunta, que tal vez le sirvieran de ayuda nemotécnica suplementaria.

Otra importante mejora sería utilizar varias pantallas para poder mostrar con amplitud preguntas con gráficos adjuntos, con código estructurado, etc., y en pantalla aparte comparar la respuesta del usuario con la verdadera. Otra pantalla con la lista de temas (incluso con apartados, aunque solo se puedan seleccionar los temas), lo que es también de gran ayuda al aprendizaje de la asignatura. Esto exigiría un gran rediseño de la app, incorporando menús de navegación, distintas 'Activity' con sus correspondientes gestores de intercambio de datos (Intent), etc.

Todo es mejorable, pero conviene no olvidar que hay que primar la eficiencia, y tal vez algunas mejoras compliquen tanto el diseño que resulten contraproducentes.

Por ejemplo, para que el usuario pudiese editar la BD de Preguntas/Respuestas y añadir/quitar o modificar lo que desee, en lugar de tener la BD original dentro de la app en /res/values/strings.xml, la tendríamos en un archivo accesible desde la app, y la gestionaríamos desde otra Activity de la app. La solución más sencilla podría ser la siguiente:

Usaríamos como BD original un archivo de texto accesible desde la app, ej. /data/data/com.example.androidflash/q.txt, con una estructura como la que se muestra en la Fig. 4 en formato CSV.

	A	B	C	D	E	F
1	clave	valor				
2	TEMA 1					
3	t1_p1	Indica cómo se denomina cada pantalla a mostrar en una aplicación.				
4	t1_r1	Activity				
5	t1_p2	Componente visual mediante el cual el usuario de alguna manera control				
6	t1_r2	Widget				
7	t1_p3	Atributos o propiedades que son necesarias para dar altura y anchura a c				
8	t1_r3	layout_width y layout_height				
9	t1_p4	Para el layout raíz, el valor de los atributos width y height es:				
10	t1_r4	match_parent				
11	t1_p5	Para los layouts o widgets internos al layout raíz los atributos width y hei				
12	t1_r5	wrap_content				
13	t1_p6	Componente no visual destinado a controlar la distribución, la posición y				
14	t1_r6	Layout				
15	t1_p7	Componente esencial que utiliza android para moverse de una pantalla (a				
16	t1_r7	Intent				
17	t1_p8	Archivo de configuración principal en formato xml donde se establecen lo				
18	t1_r8	AndroidManifest.xml				

Fig. 4. Esquema de diseño del nuevo formato de la BD.

La ventaja de esta solución es que, respetando este formato de claves para las preguntas y respuestas, bastaría modificar el método almacenar() en la app, para que tomase los valores de la clave-pregunta y la clave-respuesta desde este archivo, y no desde el interior de la app (/res/values/strings.xml).

La desventaja sería que se le exigiría al usuario tener mucho cuidado en la edición (o al programador garantizar por código la consistencia de esta BD plana, en esa segunda Activity de edición).

Otra solución sería no basarse en un archivo de texto plano, sino en uno de shared_preferences, para tener ya la estructura clave/valor. Incluso se podría usar el archivo actual, /data/data/com.example.androidflash/shared_prefs/q.xml, para ser gestionado por la segunda Activity, con lo que se podría eliminar el método almacenar().

Y un botón para poder cambiar el idioma de la interfaz, que con Android no es difícil y extendería enormemente el rango de posibles usuarios.

Bibliografía

Android. (25 de Mayo de 2023). Obtenido de wikipedia: https://es.wikipedia.org/wiki/Android

AnkiDroid. (24 de Mayo de 2023). Obtenido de play.google: https://play.google.com/store/apps/details?id=com.ichi2.anki&hl=es&gl=US

Apps. (25 de Mayo de 2023). Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_m%C3%B3vil

Documento. (23 de mayo de 2023). Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Documentos_de_Google

Duolingo. (24 de Mayo de 2023). Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Duolingo

Flashcard. (21 de mayo de 2023). Obtenido de Wikipedia: https://en.wikipedia.org/wiki/Flashcard

Google docs. (24 de Mayo de 2023). Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Documentos_de_Google

Google Play. (25 de Mayo de 2023). Obtenido de support.google: https://support.google.com/googleplay/android-developer/answer/11926878?hl=es-419

Java. (25 de Mayo de 2023). Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)

Kahoot! (23 de Mayo de 2023). Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Kahoot!

Memrise. (24 de Mayo de 2023). Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Memrise

Núcleo Linux. (25 de Mayo de 2023). Obtenido de Wikipedia: https://es.wikipedia.org/wiki/N%C3%BAcleo_Linux

Play.google. (24 de Mayo de 2023). Obtenido de play.google: https://play.google.com/store/apps/details?id=com.ichi2.anki&hl=es&gl=US

Quizizz. (21 de Mayo de 2023). Obtenido de Wikipedia: https://en.wikipedia.org/wiki/Quizizz

Quizlet. (24 de Mayo de 2023). Obtenido de Wikipedia: https://en.wikipedia.org/wiki/Quizlet

Sistema de Leitner. (25 de Mayo de 2023). Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Sistema_de_Leitner

Tarjeta didácticas. (25 de Mayo de 2023). Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Tarjeta_de_aprendizaje