



Universidad Politécnica
de Madrid



**Escuela Técnica Superior de
Ingenieros Informáticos**

Grado en Ingeniería informática

Trabajo Fin de Grado

**Aplicación Móvil de Juego de
Matemáticas para la enseñanza primaria**

Autor: Tiance Hu

Tutor: Vicente Martínez Orga

Madrid, junio de 2022

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado

Grado en Ingeniería Informática

Título: Aplicación Móvil de Juego de Matemáticas para la enseñanza primaria
Junio del 2022

Autor: Tiance Hu

Tutor:

Vicente Martínez Orga
Departamento de Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

En los últimos cincuenta años, cada año están saliendo más tecnologías avanzadas en varias áreas. En el área de la comunicación del teléfono móvil, ha surgido la tecnología 1G, 2G, 3G, 4G, y el más reciente es el 5G. Desde la aparición del primer smartphone ya ha pasado 30 años y los dispositivos móviles cada año se están utilizando por más personas y, cada vez son más jóvenes.

Ante este contexto, identificamos la oportunidad de proporcionar una aplicación que podrá aumentar las capacidades de los niños en una forma entretenida.

El objetivo de este trabajo es diseñar y desarrollar una aplicación móvil de Android sobre un juego de matemáticas para la enseñanza primaria. La aplicación tendrá el objetivo como practicar las capacidades matemáticas, de cálculos y geométricas de los niños mediante los ejercicios matemáticos de distintos niveles y distintos operadores, también se dispondrán la presentación de las figuras geométricas más comunes para el conocimiento geométrico de los niños.

Utilizaremos el Android Studio, un entorno de desarrollo integrado oficial para la plataforma Android basando en lenguaje de programación Java. Esta aplicación es la más común y dispone muchos recursos y tutoriales por internet, por lo tanto, también nos facilita el aprendizaje de esta.

El desarrollo de una aplicación consiste en varias fases, desde las definiciones de requisitos hasta la representación de esta. La representación gráfica final tendrá un entorno animado, ajustando al gusto de los niños como los tipos de letra como los colores atractivos.

Abstract

In the last fifty years, more advanced technologies are emerging every year in various areas. In the area of mobile phone communication, 1G, 2G, 3G, 4G technology has emerged, and the latest is 5G. Since the appearance of the first smartphone, 30 years have passed, and mobile devices are being used by more people every year and they are getting younger and younger.

Given this context, we identified the opportunity to provide an application that can increase children's abilities in an entertaining way.

The objective of this work is to design and develop an Android mobile application for a math game for primary education. The application will have the objective of practicing the mathematical calculation and geometric abilities of children through mathematical exercises of various levels and different operators; the presentation of the most common geometric figures for the geometric knowledge of children will also be arranged.

We will use Android Studio, an official integrated development environment for the Android platform based on the Java programming language. This application is the most common and has many resources and tutorials on the Internet; therefore, it also makes it easier for us to learn it.

The development of an application consists of several phases, from the definitions of requirements to the representation of it. The final graphic representation will have an animated environment, adjusting to the taste of children, such as fonts and attractive colours.

Índice de contenidos

1	Introducción	1
1.1	Motivación del Proyecto	1
1.2	Objetivos	1
1.2.1	Objetivos principales:	2
1.3	Estructura de la Memoria	2
2	Estado del Arte	3
2.1	Android Studio	3
2.2	Lenguaje	4
2.2.1	Java	5
2.2.2	XML	5
2.3	Diagrams.net	6
2.4	Google Drive	7
2.5	Font Awesome	7
3	Análisis y requisitos	9
3.1	Análisis de la interfaz	9
3.1.1	El tono de color	9
3.1.2	La tipografía	10
3.2	Ciclo de vida	11
3.2.1	Modelo Cascada	12
3.3	Especificación de Requisitos	12
3.3.1	Descripción del juego	12
3.3.2	Requisitos Funcionales	13
3.3.3	Requisitos no Funcionales	16
4	Desarrollo	17
4.1	Diseño	17
4.1.1	El diagrama de flujo de la navegación	17
4.1.2	Diagramas de flujo del juego	19
4.1.2.1	Diagrama de flujo del módulo de prácticas	20
4.1.3	Boceto del diseño de aplicación	20
4.1.3.1	Pantalla principal	20
4.1.3.2	Pantalla del módulo de aprendizaje	21
4.1.3.3	Pantalla del módulo de prácticas	22
4.1.3.4	Pantalla de introducción	22
4.1.3.5	Pantalla de juego	23
4.1.3.6	Pantalla de resultado	25
4.2	Implementación	25
4.2.1	Layout	25
4.2.2	Componentes de Layout	27

4.2.2.1	Button	27
4.2.2.2	ImageView.....	27
4.2.2.3	TextView	28
4.2.2.4	ViewPager2	28
4.2.3	CountDownTimer	29
4.2.4	Service.....	29
4.2.4.1	Adición de música del fondo	29
4.2.5	Fragment y FragmentStateAdapter.....	31
4.2.6	Toast	33
4.2.7	Almacenamiento de datos	35
4.2.8	Adición de colores.....	36
4.3	Estructura del código.....	37
4.3.1	Estructura de código	37
4.3.2	Estructura de los elementos de diseño	38
4.4	Pruebas de la aplicación	40
4.4.1	Pruebas unitarias.....	40
4.4.2	Pruebas del sistema.....	42
5	Resultados y Evaluación.....	44
5.1	Resultado de la aplicación.....	44
5.1.1	Captura de la pantalla de inicio	44
5.1.2	Captura de la pantalla del módulo práctica	45
5.1.3	Capturas de la pantalla de modo Ilimitado.....	45
5.1.4	Captura de la pantalla de modo Desafío	47
5.1.5	Capturas de la pantalla del módulo aprendizaje.....	48
5.2	Análisis de usuario	50
5.2.1	Prueba de accesibilidad 1	50
5.2.2	Prueba de accesibilidad 2	51
5.3	Problemas encontrados.....	52
6	Conclusiones y Líneas futuras	53
6.1	Conclusiones.....	53
6.2	Definición de líneas futuras	53
7	Análisis de Impacto.....	54
8	Bibliografía	55
9	Anexos.....	58
9.1	Índice de figuras	58
9.2	Índice de tablas	59

1 Introducción

Desde hace varios años, la educación se ha convertido en uno de los principales temas a tratar, tanto para los políticos como para los ciudadanos en general. La educación primaria es especialmente importante para el desarrollo de los niños, porque es la base de la educación en su vida, es la etapa en la que se constituye su carácter, las habilidades básicas que necesita para afrontar las diferentes situaciones que se presentan en su vida, aprenderán los valores y habilidades para su desarrollo social, etc. [1]

Entre las diversas disciplinas de la educación primaria, se destacan las matemáticas, además de ser una de las materias favoritas de los niños [2], también es una pieza clave para su desarrollo, porque les ayuda a formar su personalidad, a ser lógicos y a aprender a razonar. Las matemáticas están presentes en nuestra vida diaria constantemente. [3] Por otra parte, en el apartado académico, las matemáticas son la ciencia por excelencia al ser la base para cualquier acercamiento científico independientemente de la disciplina, ya sea en los campos de la física, la estadística o incluso la historia. Pero no solo eso, aprender esta disciplina tiene grandes beneficios para la mente, a veces desconocidos, como ayudar a tener pensamiento analítico, desarrollar la habilidad investigadora, contribuir a la agilidad mental, desarrollar la capacidad de pensamiento, fomentar la sabiduría y la curiosidad, etc. [4] [5]

1.1 Motivación del Proyecto

Mi pasión por las matemáticas ya aparecía cuando era pequeño, me interesan las matemáticas no sólo porque es una asignatura principal e importante de cualquier nivel de educación y en cualquier estado, sino lo más importante para mí es que las matemáticas me mejoran a desarrollar mi propia forma de pensar y la lógica del mundo. Y con el nacimiento de mi sobrino, he decidido hacer una aplicación móvil que servirá para él cuando entre en primaria.

Dado que las aplicaciones móviles se convierten en moda, durante los últimos años, gracias a su portabilidad, aprovechamos esta oportunidad para tener el primer contacto con Android y las aplicaciones móviles.

1.2 Objetivos

Este proyecto pretende inspirar el interés de los niños en las matemáticas, diseñar y desarrollar una aplicación móvil para la enseñanza primaria, que pueda practicar las capacidades geométricas y de cálculo de los niños mediante los ejercicios matemáticos y figuras geométricas, así como divertirse en sus tiempos ocios. Además, la aplicación proporcionará los ejercicios de varios niveles de dificultades para los niños de distintas capacidades.

1.2.1 Objetivos principales:

- Conocer y familiarizarse con el entorno de desarrollo Android Studio
- Implementación del módulo de lectura
- Implementación del módulo de juego
- Ajustar diseño gráfico ajustando al gusto de los niños
- Adición de músicas y sonidos

1.3 Estructura de la Memoria

A continuación, se explica la estructura de esta memoria, los distintos capítulos que tiene y una breve descripción.

Introducción: Se presentará una introducción del trabajo, la motivación del proyecto y el objetivo de este.

Estado del Arte: Se explicará las distintas tecnologías utilizadas o empleadas durante el desarrollo del proyecto, así como el lenguaje de programación, el entorno utilizado, la herramienta de desarrollo, las fuentes de elementos multimedia, etc. [6]

Análisis y Requisitos: Se introducirá una breve descripción del proyecto, y se realizará los análisis de los requisitos del producto.

Desarrollo: En esta sección se discutirá el diseño del producto, así como su implementación, los distintos elementos utilizados para la construcción de la interfaz gráfica. Y al final de la sección se explicará cómo se repartía el trabajo, las tareas realizadas y un diagrama de Gantt seguida.

Pruebas y Evaluación: Se mostrará el resultado final de la aplicación móvil y se explicará el análisis de las pruebas de accesibilidad realizadas por los niños y los problemas detectados durante la realización de prueba.

Conclusión y Líneas futuras: En este apartado se presentará las conclusiones tras el desarrollo de todo el proyecto y las propuestas de mejora de la aplicación.

Análisis de impacto: En este capítulo se realizará un análisis del impacto potencial de los resultados obtenidos durante la realización del TFG en diferentes contextos, y se destacarán los beneficios esperados, así como los posibles efectos perjudiciales.

Por último, un capítulo con la bibliografía empleada para la redacción este proyecto, diagramas de clase completo, el índice de figuras y el índice de tablas.

2 Estado del Arte

En este capítulo se van a explicar detalladamente las herramientas, los artículos, los entornos, las fuentes de recursos que se han necesitado o utilizado para el desarrollo del proyecto.

2.1 Android Studio

Android Studio es un entorno de desarrollo integrado (IDE) oficial creado por Google, y está especializado para desarrollar aplicaciones de Android. Se basa en IntelliJ IDEA y con él se puede desarrollar aplicaciones para teléfonos Android, Wear OS by Google, Android TV, Android Auto, Android Things, etc. Proporcionando todas las funcionalidades que se necesita cada fase del desarrollo. [7]



Figura 1. El logo de Android Studio

Además, Android Studio posee las siguientes funcionalidades que aumentan la productividad cuando se desarrolla aplicaciones para Android, como las siguientes:

- Un sistema de compilación flexible basado en Gradle.
- Un emulador rápido y cargado de funciones.
- Un entorno unificado donde se puede desarrollar para todos los dispositivos Android.
- Aplicación de cambios para insertar cambios de código y recursos a la app en ejecución sin reiniciarla.
- Integración con GitHub y plantillas de código para ayudar a compilar funciones de apps comunes y también importar código de muestra.
- Variedad de marcos de trabajo y herramientas de prueba.
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones, entre otros.
- Compatibilidad con C++ y NDK.

- Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine.

Entre estas funcionalidades de Android Studio, se destaca su emulador que permite emular un dispositivo Android en una computadora, por lo que puede probar la aplicación en una variedad de dispositivos Android y niveles de API de Android sin tener los dispositivos físicos. El emulador proporciona casi toda la funcionalidad de un dispositivo Android real. Se puede simular llamadas entrantes, mensajes de texto, especificar la ubicación del dispositivo, simular diferentes velocidades de Internet, simular la rotación y otros sensores de hardware, etc. [8]

La aplicación se ejecutará en el Android Emulador que dispone una variedad de dispositivos y niveles de API de Android para la simulación en una computadora y sin necesidad de tener los dispositivos físicos.

Para ello, se necesita crear primero un dispositivo virtual de Android (ADV) seleccionando el dispositivo y el Api. Posteriormente cuando se ejecuta una aplicación, Android Studio instalará la aplicación en el dispositivo y la ejecutará iniciando el emulador.

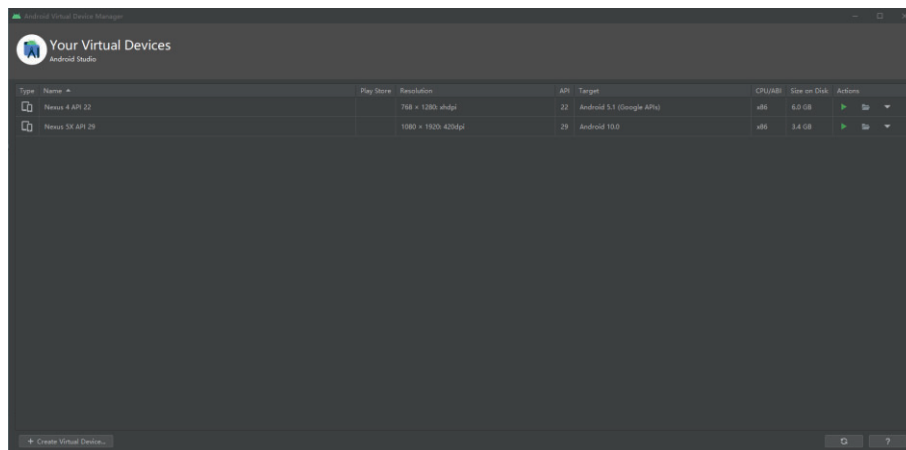


Figura 2. Administrador de dispositivos virtuales de Android

2.2 Lenguaje

Durante el desarrollo de trabajo, se utiliza principalmente dos lenguajes para el desarrollo de aplicación. Por un lado, el lenguaje de programación Java para el funcionamiento del programa, y por otro lado, el lenguaje de marcado XML para definir las especificaciones de la representación de interfaz gráfica.

2.2.1 Java

Java es un lenguaje de programación de propósito general, tipado, orientado a objetos creado por Sun Microsystems, permite el desarrollo de cualquier tipo de programa. Fue diseñado para ser multiplataforma y poder ser empleado el mismo programa en diversos sistemas operativos. [9]



Figura 3. El logo de Java

Java es un lenguaje simple, seguro y sólido, y fácil de aprender, se ha dedicado utilizar este lenguaje no sólo porque es el lenguaje más utilizado a lo largo de la carrera, y se ha llevado más experiencia en ello, sino también por su gran variedad de librerías que ayudarían considerablemente en cuanto a la hora del desarrollo de la aplicación.

2.2.2 XML

XML es un lenguaje de marcado similar al HTML, utilizado para describir datos. Las etiquetas XML no están predefinidas en XML, por lo cual se debe definir las propias etiquetas. XML está caracterizado por su buena legibilidad tanto por los humanos como por las máquinas, además, en cuanto al desarrollo, es escalable y muy fácil en aprendizaje. En Android, se usa XML para diseñar la representación de la interfaz gráfica, y gracias a su ligereza, no hace que el diseño sea pesado. [10]



Figura 4. XML

En Android Studio, los archivos XML se utiliza para varios propósitos diferentes [11]:

- **Layout XML:** Los archivos para el propósito de diseño se utilizan para definir la interfaz gráfica de usuario real de la aplicación, contiene todos los elementos de vista o las herramientas que se quieren utilizar en la aplicación como, por ejemplo, los TextView, los botones y otros elementos de la interfaz de usuario que se explicará con más detalle en el capítulo de desarrollo.
- **Manifest XML:** Este archivo XML se utiliza para definir todos los componentes de nuestra aplicación. En ello incluye los nombres de paquetes de la aplicación, las actividades, receptores, servicios y los permisos que se necesita en la aplicación.
- **String XML:** En este archivo se podrá definir las cadenas de caracteres que se utilizará frecuentemente en el desarrollo de aplicación con finalidad de mejorar la reutilización del código.
- **Styles XML:** Este XML se utiliza para definir los diferentes estilos y temas personalizados de la aplicación.
- **Color XML:** En este fichero se define los códigos de color que se utiliza en la aplicación.

2.3 Diagrams.net

Diagrams.net (anteriormente llamado draw.io), es una aplicación web gratuita y de código abierto con finalidad de creación y edición de cualquier tipo de diagramas en cualquier navegador web sin necesidad de instalación.

Con ella se puede crear y editar una gran variedad de diagramas como: diagramas entidad-relación, diagramas de flujo, diagramas UML, diagramas de procesos, etc. Y los resultados creados se guardarán directamente en el Google Drive. [12]

El uso principal de esta herramienta en el desarrollo del proyecto es crear las diagramas de clases, diagramas de flujos y los diseños.

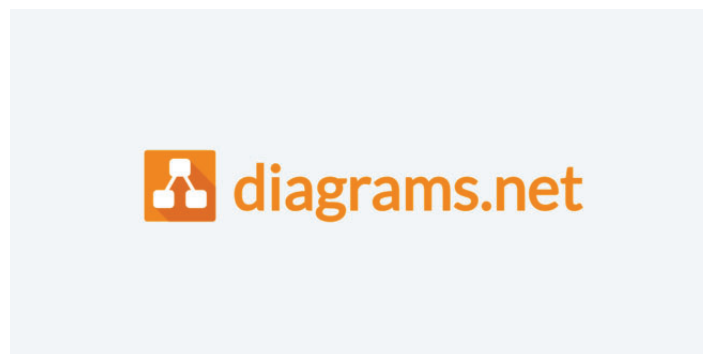


Figura 5. El logo de diagrams.net

2.4 Google Drive

Google Drive es un servicio de almacenamiento de datos en internet que provee Google en su versión gratuita con una capacidad de almacenamiento de 15GB. Para el desarrollo de proyecto se elige esta plataforma por su flexibilidad, fácil de utilizar y relacionado con Diagrams.net, es decir, se puede acceder a ello desde cualquier dispositivo que nos permite consultar, editar, buscar los archivos en cualquier sitio y en cualquier dispositivo. Además, se puede crear, diseñar y editar directamente en esa plataforma las diagramas de Diagrams.net. [13]

El uso principal de esta herramienta en el desarrollo del proyecto es la gestión de versiones de archivo, guardar la copia de seguridad de la memoria y crear las diagramas de Diagrams.net.



Figura 6. El logo de Google Drive

2.5 Font Awesome

Font Awesome es un repositorio impresionante de iconos vectoriales y estilo CSS utilizado para sustituir imágenes de iconos comunes por gráficos vectoriales convertidos en fuentes. FontAwesome nos ayudan a ganar tiempo en la implementación de aplicación. [14]



Figura 7. El logo de Font Awesome

Font Awesome dispone de una versión gratuita y otra de pago. En el desarrollo del proyecto se usará la versión gratuita que dispone más de 2000 iconos, y además se ha elegido esta fuente también por su licencia generosa, se puede utilizar sus contenidos de cualquier uso incluyendo el uso comercial.

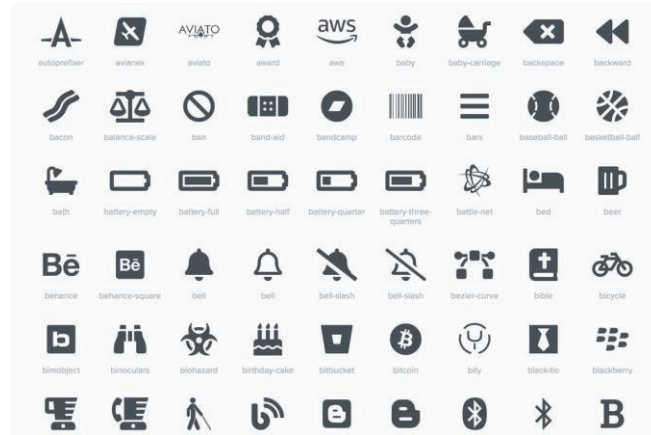


Figura 8. Ejemplo de la galería de iconos FontAwesome

3 Análisis y requisitos

En este capítulo se presenta el análisis de la interfaz gráfica, el tipo de ciclo de vida que se aplica en desarrollo de aplicación y las especificaciones de los requisitos que se debe cumplir la aplicación.

3.1 Análisis de la interfaz

En este apartado se estudia la mejor manera de diseñar una interfaz gráfica de aplicación para los niños, prestando especial atención al tono de color y texto que se aplica.

3.1.1 El tono de color

Los colores tienen un gran influencia en los niños, pueden cambiar emociones, despertar la motivación, darnos identidad, y familiarizar la comunicación con los demás. El color también afecta en gran medida el comportamiento, estado de ánimo y aprendizaje de los niños. [15]

La ciencia está de acuerdo en que los colores tienen un efecto en el estado de ánimo de las personas. Y que también pueden mejorar la concentración, reducir o aumentar la agresividad, ayudar a conciliar el sueño o estimular la memoria.

El tono de color de aplicación que se aplica en la interfaz gráfica es de un tono amarillo ligero con botones de azul claro y textos de verde claro por varias razones [16] [17]:

- Los niños pequeños suelen preferir el azul, verde o amarillo, mientras que las niñas prefieren el coral, el lavanda y el rosa suave.
- El amarillo es un color asociado a la estimulación intelectual. Al igual que el color rojo, el amarillo es un color muy vital. Además, por el optimismo que transmite, el amarillo es muy beneficioso para estimular la atención de los niños y es bastante útil para niños que sufren de depresión.
- El azul es un color relajante que transmite profundidad y relajación. Con este color, es más fácil que los niños se concentren a la hora de estudiar o realizar las tareas cotidianas. Genial para los niños nerviosos o aquellos que les cueste tomar el hábito del estudio.
- El verde es un color excelente para transmitir armonía y calmar el sistema nervioso. Puede ayudar a mejorar la habilidades de lectura, tanto la velocidad como la comprensión lectora.
- Los colores claros y ligeros inspiran ligereza, se percibe como aireada, ligera, tranquilo y agradable para no crear el sentimiento de fatigado y estancado.

El poder de los colores sobre los niños		guiainfantil.com
Color	Qué transmite	Beneficioso para...
Blanco	Pureza, calma y orden visual	Incentiva la creatividad
Azul	Calma, serenidad	Mejora el sueño. Bueno para niños nerviosos
Rojo	Energía, vitalidad	Ayuda en niños más tímidos
Amarillo	Positivismo, energía	Estimula la concentración. Bueno para niños con depresión
Verde	Equilibrio y calma	Mejora la capacidad lectora
Naranja	Energía y positivismo	Estimula la comunicación
Morado	Tranquilidad y misterio	Potencia la intuición

Figura 9. Las actitudes que transmiten los colores a los niños [17]

3.1.2 La tipografía

El tipo de letra también es significativo para la lectura de los niños, deben ser lo más legible y familiar posible para los niños. Tradicionalmente se parte de letras enlazadas que parecen ayudar la escritura, ya que el hecho de que las letras estén enlazadas entre sí permite una mayor fluidez en la escritura. Además, evita la típica confusión entre letras simétricas (d – b, p – q) ya que este tipo de letra no tienen esta simetría. [18]

Aunque cada vez son más colegios que empiezan con la letra de impreso, ya que es el que más aparece en los libros y dispositivos móviles que los niños ya utilizan desde pequeños.

En la aplicación, se decide utilizar el tipo de letra Comic Sans en general y la tipografía manuscrita Escolar para la lectura. Comic Sans es una fuente habitualmente asociado a textos infantiles, tiene ventajas en su claridad, facilidad de leer y la mayoría de los niños ya ha tenido contacto frecuentemente con este tipo de letra. [18]

Comic Sans MS
AaBbCcDdEeFfGgHhIi
JjKkLlMmNnOoPpQqRr
SsTtUuVvWwXxYyZz
1234567890!@#\$%^&
*()-=[\];',./_+{}:"<>?\`|~

Figura 10. Las letras de tipografía Comic Sans MS

Se decide utilizar una tipografía de letras manuscritas para la lectura porque en España, las letras manuscritas aparecen mucho en las lecturas de colegio y son leídos por miles y miles de niños, por lo tanto, están también muy familiarizados con esta tipografía. [19]

Q w e r t y u i o p
a s d f g h j k l ñ z
x c v b n mamá
guapa murciélago,
león

Figura 11. Las letras de tipografía Escolar

3.2 Ciclo de vida

El ciclo de vida del software es indudablemente un proceso imprescindible para el desarrollo del software si se quiere conseguir un producto de manera exitosa. No se trata de un proceso para conseguir simplemente un producto de alta calidad y que cumpla todos los requerimientos de los clientes, sino también debe terminarse en tiempo y cumplirse los costes estimados.

Existen distintos modelos de ciclo de vida, y los más comunes son siguientes [20]:

- Cascada
- Incremental
- Evolutivo
- Iterativo
- Prototipo
- Espiral
- En V



Figura 12. Ciclo de vida del software

3.2.1 Modelo Cascada

El ciclo de vida que se aplica en el desarrollo de aplicación es el modelo cascada, es un modelo de ciclo de vida clásico que engloba todas las fases del desarrollo de software, desde la fase de especificación de requisito, pasando por la fase de diseño, la fase de implementación, la fase de integración y pruebas, hasta la última fase que es la de mantenimiento. Además, este modelo adquiere una estructura bastante simple debido a sus fases de proyecto claramente diferenciadas. [21]

La razón por la que se eligió a este modelo no es solo porque es el modelo más clásico, sino también porque es el modelo más usado durante la carrera, que ya tiene una cierta base con ello.

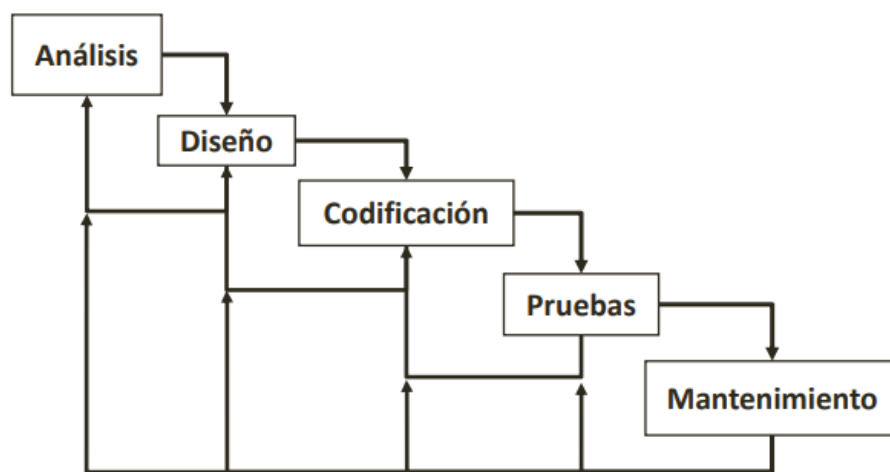


Figura 13. Ciclo de vida cascada

3.3 Especificación de Requisitos

En este apartado se define todas las necesidades y los requisitos que se debe cumplir la aplicación tanto los funcionales como los no funcionales.

3.3.1 Descripción del juego

El juego tendrá dos módulos, un módulo consiste en el juego matemático que dispondrá dos modos de juego y para cada modo tres niveles de dificultades para niños de diferentes niveles y distintos cursos de primaria, se pueden ajustar el nivel de dificultad fácilmente. Para cada modo de juego de cada nivel se guardará el mejor récord de la partida y se visualiza en la pantalla.

El juego se basa en los ejercicios de operaciones aritméticas, y dependiendo del nivel que se ajusta, apareciendo preguntas de distintas operaciones aritméticas y más número de cifras: en el nivel fácil solamente están la suma y la resta de una cifra, y en los otros niveles más complicados aparecerá operaciones como

la multiplicación y la división, y como máximo operaciones de tres cifras. Se dispondrán cuatro botones de respuesta para elegir, en los cuales solamente uno de ellos es correcto. También para cada partida, el jugador tiene tres vidas, al contestar una respuesta errónea, se resta una de vida y cuando se agota la vida, finaliza la partida.

En el modo 1, llamado el Modo Ilimitado, tiene infinitas preguntas y hasta que no se pierda las tres vidas no se va a finalizar la partida. Con cada pregunta el jugador tendrá 15 segundos para contestar y cuando se finaliza ese tiempo, se restará una vida y empezará la nueva pregunta en caso de que el jugador haya más de una vida. Para cada respuesta correcta se consigue 1000 puntos y al finalizar la partida se visualizará los puntos y los guardará como el nuevo récord en caso de que supere la mejor nota que ha tenido.

En el modo 2, llamado el Modo Desafío, consiste en veinte preguntas de operación aritmética, el jugador debe finalizar lo antes posible esas preguntas sin perder todas sus vidas. Al terminar todas las preguntas, se considera como éxito en caso de que el jugador disponga al menos de una vida y se mostrará los tiempos que se ha consumido y se guardará este resultado si se trata de un nuevo récord. Cada vida perdida se sumará 30 segundos al resultado final en el final de la partida.

En ambos modos, se dispone al principio de la partida una introducción sobre el modo de juego para que el jugador sepa cómo funciona ese modo.

Por la otra parte está el módulo de aprendizaje, este módulo tiene objetivo como enseñar a los niños las figuras geométricas más utilizados junto con sus fórmula de área y perímetro. También se dispone en esa parte las chuletas de operaciones aritméticas para recordar o enseñar a los niños cómo se hacen las operaciones aritméticas, las explicaciones tienen que ser acompañadas por imágenes.

3.3.2 Requisitos Funcionales

Tabla 1. R1. Seleccionar el módulo de aplicación.

Requisito 1: Seleccionar el módulo de aplicación
Descripción: El usuario puede acceder al módulo de aprendizaje o al módulo de práctica mediante un clic en el botón.

Tabla 1. R1. Seleccionar el módulo de aplicación

Tabla 2. R2. Cambiar de una chuleta a la otra mediante clic.

Requisito 2: Cambiar de una chuleta a la otra mediante clic
Descripción: En el módulo de aprendizaje, el usuario puede cambiar de la sección actual a la otra mediante un clic en la sección que le interese.

Tabla 2. R2. Cambiar de una chuleta a la otra mediante clic

Tabla 3. R3. Cambiar de una chuleta a la otra mediante deslizamiento.

Requisito 3: Cambiar de una chuleta a la otra mediante deslizamiento
Descripción: En el módulo de aprendizaje, el usuario puede cambiar de la sección actual a la siguiente o la anterior deslizando el dedo hacia arriba o abajo.

Tabla 3. R3. Cambiar de una chuleta a la otra mediante deslizamiento

Tabla 4. R4. Silenciar o reproducir la música de fondo.

Requisito 4: Silenciar o reproducir la música de fondo
Descripción: El usuario puede silenciar o reproducir la música de fondo haciendo clic en el icono de altavoz.

Tabla 4. Silenciar o reproducir la música de fondo

Tabla 5. R5. Seleccionar el nivel de dificultad.

Requisito 5: Seleccionar el nivel de dificultad
Descripción: En el módulo de práctica, el usuario puede elegir uno de los tres niveles de dificultad que se quiere seguir en la partida mediante un clic en el botón.

Tabla 5. R5. Seleccionar el nivel de dificultad

Tabla 6. R6. Seleccionar el modo de juego.

Requisito 6: Seleccionar el modo de juego
Descripción: En el módulo de práctica, el usuario puede elegir uno de los dos modos de juego de la partida mediante un clic en el botón.

Tabla 6. R6. Seleccionar el modo de juego

Tabla 7. R7. Mostrar el récord de cada modo y en cada nivel.

Requisito 7: Mostrar el récord de cada modo y en cada nivel
Descripción: En el módulo de práctica, el usuario puede visualizar el mejor récord que ha tenido para cada nivel de cada modo en la parte superior de los botones de juego.

Tabla 7. R7. Mostrar el récord de cada modo y en cada nivel

Tabla 8. R8. Iniciar una partida.

Requisito 8: Iniciar una partida
Descripción: En el módulo de práctica, el usuario puede iniciar una partida haciendo clic en el botón de modo de juego.

Tabla 8. R8. Iniciar una partida

Tabla 9. R9. Mostrar la introducción de cada modo.

Requisito 9: Mostrar la introducción de cada modo
Descripción: Al iniciar la partida de cada modo, se dispondrá de una ventana con la explicación del modo juego que ha elegido el usuario, al dar un botón inferior, se apaga la ventana y se empieza la partida.

Tabla 9. R9. Mostrar la introducción de cada modo

Tabla 10. R10. Seleccionar la respuesta de una pregunta.

Requisito 10: Seleccionar la respuesta de una pregunta
Descripción: El usuario para contestar a la pregunta, selecciona una de las cuatro respuestas mediante un clic en uno de los cuatro botones. Una correcta selección sumará puntos en caso de modo Ilimitado, y una incorrecta respuesta restará una vida del jugador.

Tabla 10. R10. Seleccionar la respuesta de una pregunta

Tabla 11. R11. Mostrar los puntos que lleva durante una partida.

Requisito 11: Mostrar los puntos que lleva durante una partida
Descripción: En modo Ilimitado, se debe mostrar la calificación que lleva hasta actual, y en modo desafío, se debe mostrar el tiempo consumido de la partida.

Tabla 11. R11. Mostrar los puntos que lleva durante una partida

Tabla 12. R12. Mostrar el tiempo restante para la pregunta.

Requisito 12: Mostrar el tiempo restante para la pregunta
Descripción: En modo desafío, se muestra el tiempo que queda para contestar el jugador a la pregunta.

Tabla 12. R12. Mostrar el tiempo restante para la pregunta

Tabla 13. R13. Mostrar las preguntas restantes.

Requisito 13: Mostrar las preguntas que quedan en el modo Desafío
Descripción: En modo desafío, se debe mostrar las preguntas que quedan para finalizar la partida.

Tabla 13. R13. Mostrar las preguntas restantes

Tabla 14. R14. Mostrar las vidas restantes.

Requisito 14: Mostrar las vidas restantes
Descripción: Durante la partida, se debe mostrar las vidas que tiene el jugador.

Tabla 14. R14. Mostrar las vidas restantes

3.3.3 Requisitos no Funcionales

Tabla 15. R1. Legibilidad.

Requisito 1: Legibilidad
Descripción: Todos los botones, textos, figuras deben ser fácil de leer y fácil de entender.

Tabla 15. R1. Legibilidad

Tabla 16. R2. Usabilidad.

Requisito 2: Usabilidad
Descripción: Deben disponer en cualquier momento de un botón de salida a la pestaña anterior, el diseño de la interfaz gráfica debe ser bastante intuitivo y que sea fácil de manejar.

Tabla 16. R2. Usabilidad

4 Desarrollo

Este capítulo está dedicado a describir el desarrollo del proyecto realizado, consiste principalmente tres apartados, el diseño de la interfaz gráfica de aplicación, la implementación de la aplicación, tanto las funcionalidades como la interfaz gráfica.

4.1 Diseño

En este apartado se presenta el diseño del flujo entre las pantallas de la aplicación explicando un poco las pantallas también, el diagrama de flujo de los dos modos de juego y para terminar esta sección el boceto temporal de pantallas de la aplicación.

4.1.1 El diagrama de flujo de la navegación

Un diagrama de flujo de navegación es un diagrama que plasma la estructura de navegación de un entorno, indicando las pantallas principales de una aplicación en una forma jerárquica y secuencial. El diagrama permite una navegación preliminar, donde se puede guiar al usuario a lo largo de la ruta que tomará al utilizarlo, que le mostrará las conexiones a las diferentes pantallas que pueda tener. Es una forma rápida y directa de transmitir informaciones. [22]



Figura 14. Ejemplo de un flujo de navegación

Para modularizar las dos partes principales, se decide dividir los dos módulos en la pantalla de inicio, como se muestra en la figura 14, desde la pantalla de inicio se puede acceder a los dos módulos diferentes.

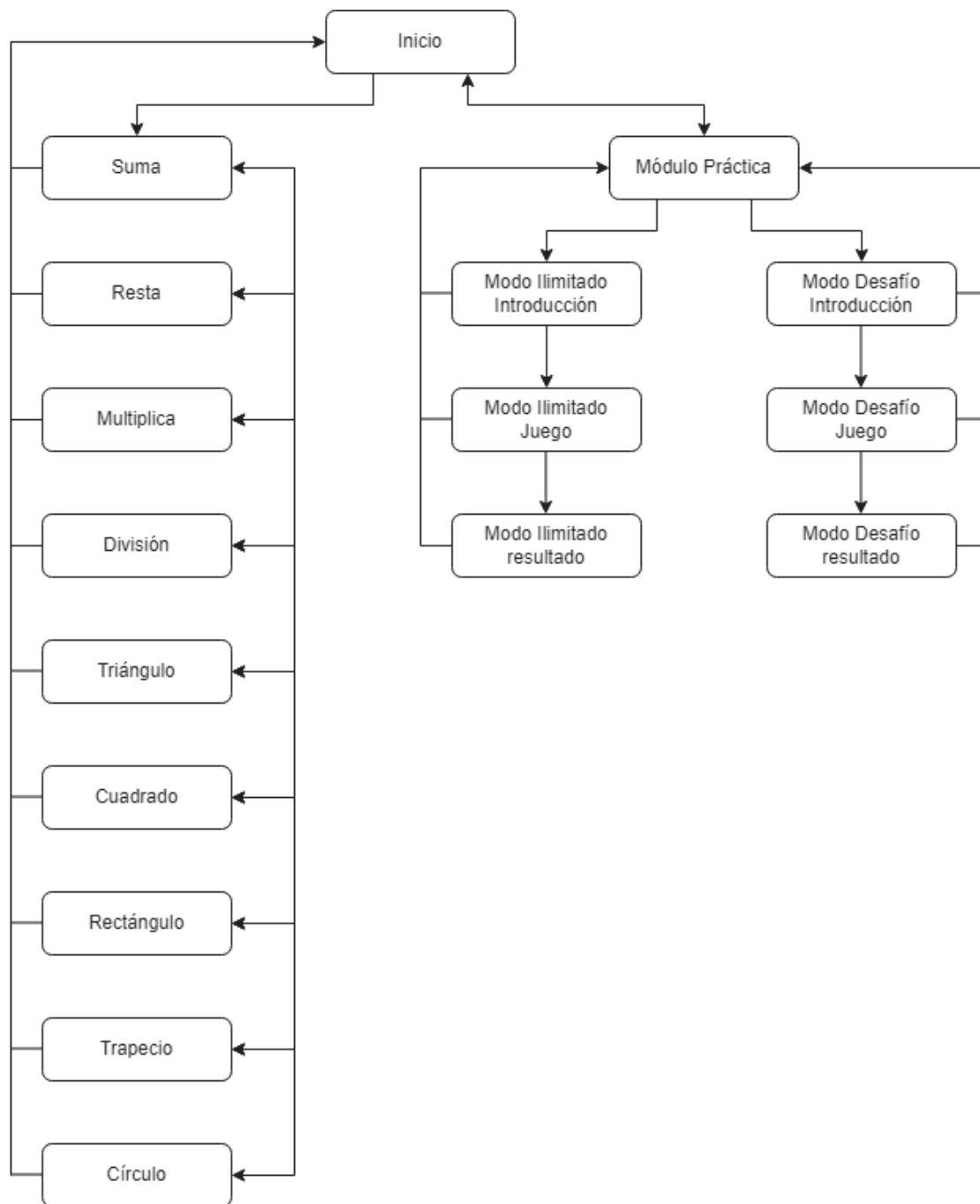


Figura 15. El flujo de la navegación de pantalla

Al acceder al módulo de aprendizaje, la pantalla por defecto es la pantalla de la operación suma, ya que es la primera operación que vemos en nuestra infancia, y se trata de una operación más básica. De la pantalla de operación suma, se puede cambiar a otras cualquier pantalla de aprendizaje mediante clic en una pestaña situada en la parte izquierda de pantalla, o bien deslizando hacia arriba o abajo para pasar a la siguiente pantalla de aprendizaje o a la pantalla anterior.

En cualquiera pantalla de aprendizaje se puede salir perfectamente del módulo de aprendizaje mediante un clic en el botón de salida, y tras dar ese botón, se volverá a la pantalla del inicio.

Para el módulo de prácticas, también se accede desde la pantalla de inicio, una vez entrado a la pantalla de prácticas, se verá los tres niveles de dificultades que son: nivel “Fácil”, nivel “Difícil” y nivel “Infierno” siendo el más fácil el nivel “Fácil”, y el nivel más complicado es el nivel “Infierno”. Además, se dispone de dos botones para acceder a los dos modos distintos de juego, mostrando para cada uno de ellos el récord mejor que ha tenido el jugador, a principio será 0 puntos y 0 segundos.

Cada modo consistirá principalmente en tres pantallas: la primera es la pantalla de introducción donde se muestra en la ventana una tutorial en forma de texto que se explica el funcionamiento del modo de juego. La segunda pantalla se trata del juego principal, básicamente son cuatro botones de respuesta y el enunciado de la pregunta, y dependiendo del modo de juego se mostrará el cronometrador y las preguntas restantes o la calificación y el tiempo restante para la pregunta. La última pantalla es simplemente un botón donde muestra la calificación que ha tenido en la partida o el tiempo consumido en la partida y al dar el botón, se vuelve a la pantalla de prácticas.

4.1.2 Diagramas de flujo del juego

Un diagrama de flujo es un diagrama que representa todos los pasos de un proceso, sistema o algoritmo informático mediante unos símbolos geométricos y unas flechas de conexión. Este diagrama ayudará a transformar un proceso complejo de variedad de campo a un diagrama claro y fácil de comprender. Además, su uso no requiere conocimientos técnicos y cualquiera puede utilizar sin problemas. [23]

Los diagramas de flujo pueden enfocarse en funciones más específicas y profundas como diagramas de flujo de procesos, mapa de procesos, etc. Y están relacionado también con otros diagramas populares como diagramas de flujo de datos y los diagramas de UML. [23]

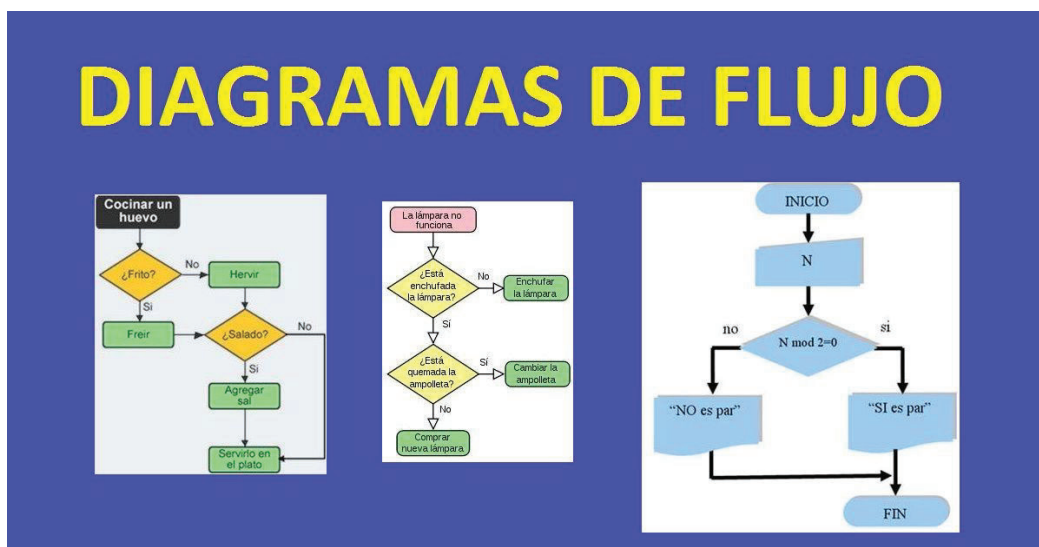


Figura 16. Ejemplos de diagramas de flujo

En esta sección se presenta el flujo de los dos modos de juego mediante un diagrama de flujo.

4.1.2.1 Diagrama de flujo del módulo de prácticas

A continuación, se muestra el diagrama de flujo del módulo de prácticas, detallando lo más posible en cada decisión.

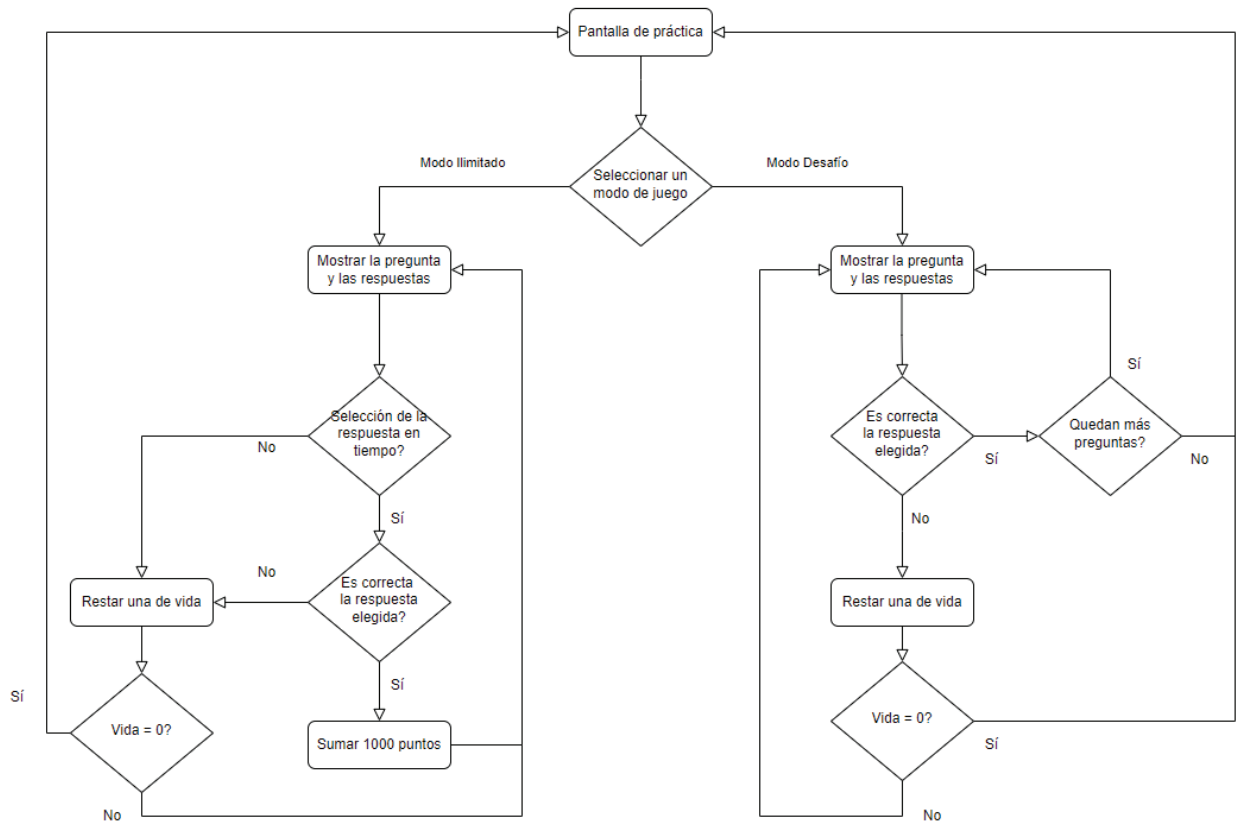


Figura 17. Diagrama de flujo del sistema de juego

4.1.3 Boceto del diseño de aplicación

En este apartado se muestra un boceto temporal de las pantallas de aplicación, el objetivo de este boceto es tener claro en cuanto a la fase de implementación cómo se debe implementar la interfaz gráfica.

4.1.3.1 Pantalla principal

En esta pantalla se debe disponer de dos botones, uno para entrar al módulo de aprendizaje y el otro para acceder al módulo de práctica. Además, también debe mostrar el nombre de la aplicación y un icono para silenciar o reproducir la música del fondo.



Figura 18. Boceto de la pantalla principal

4.1.3.2 Pantalla del módulo de aprendizaje

En esta pantalla se debe disponer una barra de menú situada en la parte izquierda de la pantalla para que puedan ser elegidos por el usuario, y la parte superior por la izquierda de la pantalla se debe disponer de dos iconos, uno para volver a la pantalla anterior y otra para silenciar la música del fondo. El contenido núcleo de la pantalla indudablemente es la chuleta y su correspondiente figura.

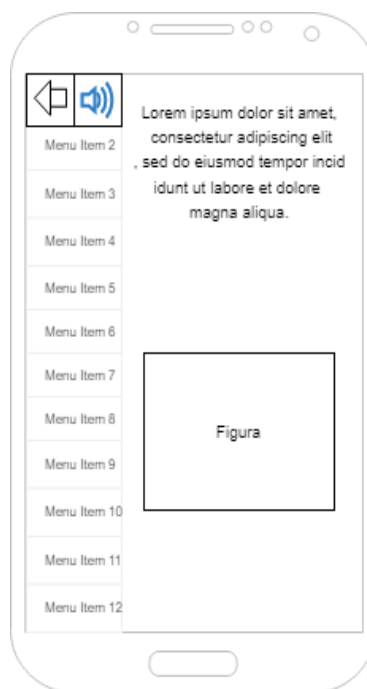


Figura 19. Boceto de la pantalla de aprendizaje

4.1.3.3 Pantalla del módulo de prácticas

En esta pantalla se presenta la pantalla donde se accede al juego de los dos modos. Dispone tres botones de niveles para que el usuario seleccione, si uno de los botones está presionado, los otros dos no pueden estar presionados lógicamente. También debe mostrar el récord de cada modo en esta pantalla. Además, se debe también disponer dos iconos como la pantalla de la sección 4.1.3.3, uno para volver a la pantalla principal y el otro para silenciar la música del fondo.

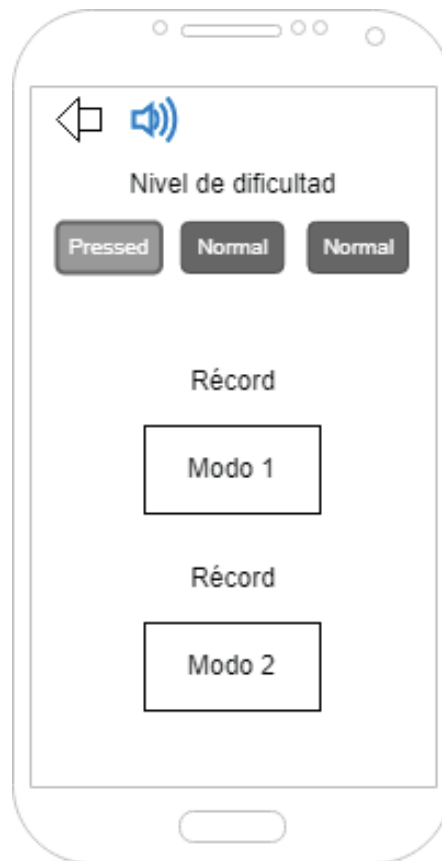


Figura 20. Boceto de la pantalla de práctica

4.1.3.4 Pantalla de introducción

En esta pantalla debe proporcionarse una ventana con la introducción del modo de juego elegido previamente por usuario, en esa ventana debe tener un botón para cerrar dicha ventana y empezar la partida. En esta pantalla también debe disponer de dos iconos, uno para volver a la pantalla de práctica y el otro icono para silenciar o reproducir la música del fondo como las pantallas mencionadas anteriormente.



Figura 21. Boceto de la pantalla de introducción del modo de juego

4.1.3.5 Pantalla de juego

Esta pantalla es la pantalla núcleo de este módulo, en ella se debe mostrar la pregunta y las respuestas. Además, también se visualiza las tres vidas que tiene el jugador a principio de una partida en la parte inferior a la izquierda de la pantalla.

Dependiendo del modo de juego:

Modo Ilimitado: Encima de la pregunta debe mostrar el tiempo restante para dicha pregunta, una vez el tiempo se acaba, se resta una de vida del jugador. Y cuando se agotan las vidas, pasa a la pantalla de resultado.

El jugador dispone cuatro respuestas en forma de botón para seleccionar, si la respuesta se trata de una respuesta errónea, también se resta una vida, y si fuese la respuesta correcta, se suma 1000 puntos y se visualizará en la parte superior a la derecha de la pantalla los puntos que lleva el jugador durante la misma partida.

Modo Desafío: La pantalla de este modo es parecida a la de modo ilimitado, también dispone una pregunta y cuatro respuestas, y se muestra las preguntas restantes en vez del tiempo restante, y también ha sustituido los puntos durante la partida por el tiempo consumido en la partida.

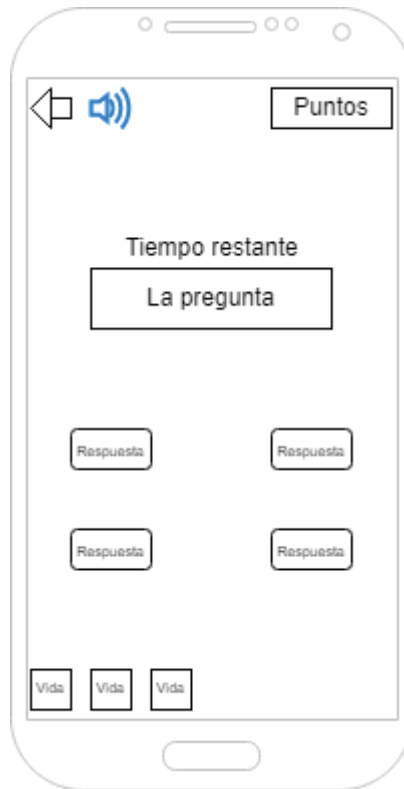


Figura 22. Boceto de la pantalla de modo Ilimitado

A continuación, se muestra el boceto de la pantalla de modo desafío.

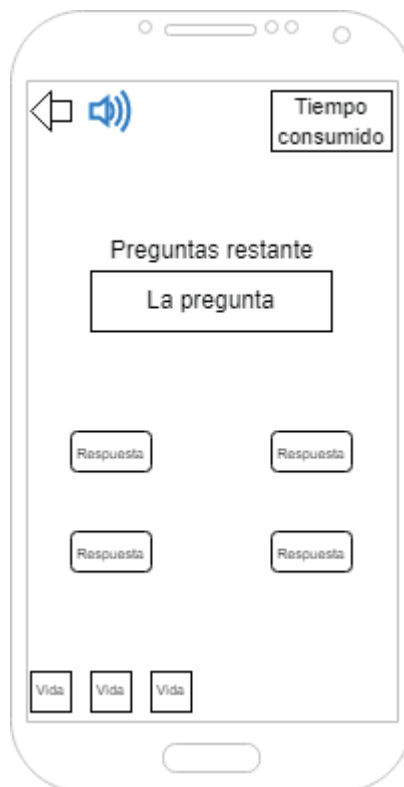


Figura 23. Boceto de la pantalla de modo Desafío

4.1.3.6 Pantalla de resultado

Esta pantalla es simplemente destinada para mostrar al jugador la información de los puntos ganados o el tiempo consumido de la partida. Una vez el jugador haya dado al botón, se vuelve a la pantalla de práctica.

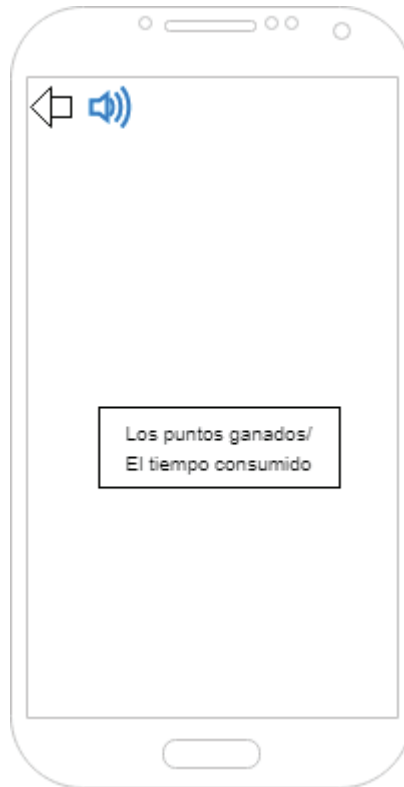


Figura 24. Boceto de la pantalla de resultado

4.2 Implementación

En este apartado se explica la implementación de la aplicación incluyendo los componentes del diseño, las librerías, técnicas y servicios utilizados para los diferentes módulos de aplicación, como el sistema de juego, módulo de aprendizaje, servicio de música del fondo, etc.

4.2.1 Layout

En Android, el Layout es un contenedor utilizado para controlar el comportamiento de los elementos visuales de diseño de la aplicación y sus posiciones incluso un Layout se coloca dentro del otro Layout. Los Layout utilizados en la aplicación son el LinearLayout y el ConstraintLayout.

El LinearLayout es el Layout más simple y utilizado, lo que hace este diseño es alinear los componentes en una dirección, ya sea horizontal o vertical utilizando el atributo android:orientation. [24]

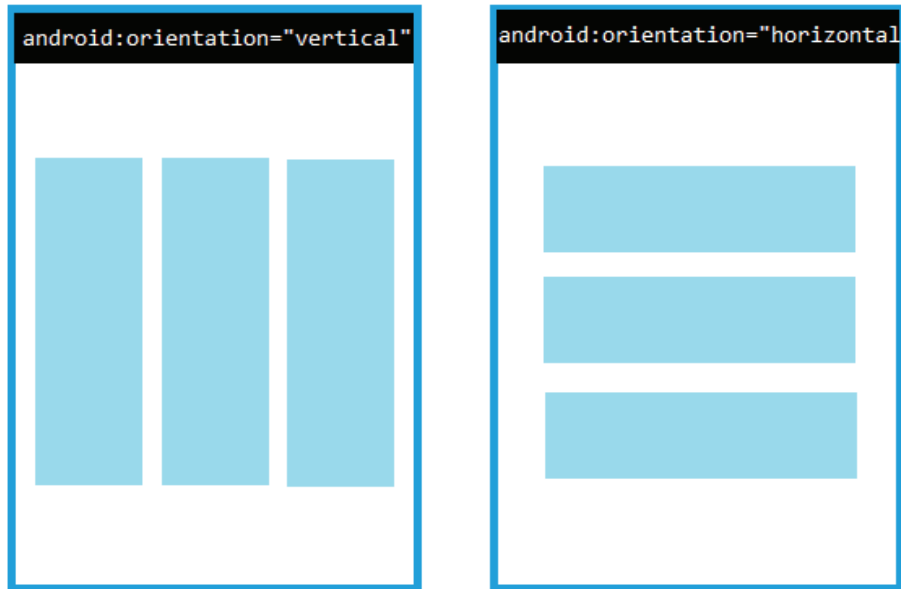


Figura 25. Ejemplo de LinearLayout

El ConstraintLayout es el diseño principal que se utiliza en la aplicación, porque este Layout reduce la complejidad del diseño en cuanto un diseño tenga muchos elementos, los organiza de una manera jerárquica y de vista plana. Este diseño también es muy flexible en adaptación, los elementos en este diseño son restringidos tanto su posición como su tamaño, por lo que se adapta mejor a diferentes modelos. [25]

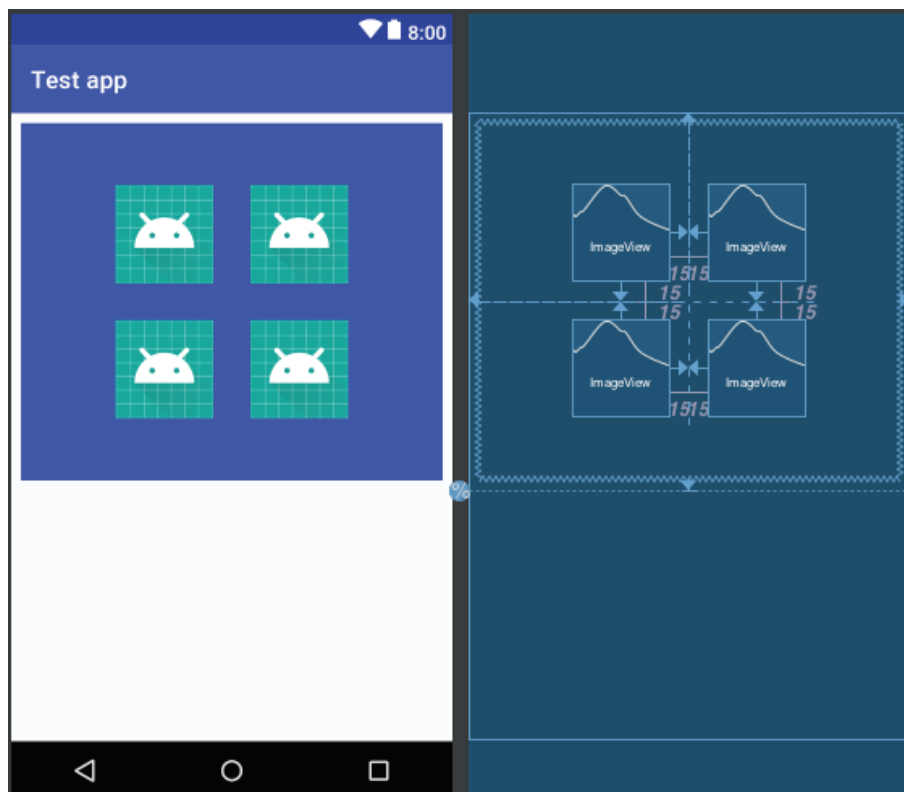


Figura 26. Ejemplo de ConstraintLayout

4.2.2 Componentes de Layout

En esta sección se explica los componentes del diseño utilizados en la implementación de la interfaz gráfica.

4.2.2.1 Button

La función principal del componente Button es responder a eventos de clic, en nuestra aplicación se utiliza para ir de una pantalla a la otra, guardar las respuestas seleccionadas por el usuario durante una partida de juego. Para implementar dichas funciones, se destaca el atributo `android:onClick` que es un atributo en la que se maneja hacia qué método se debe llamar cuando se hace un clic sobre el botón. [26] [27]

A continuación, se muestra un botón de la aplicación para navegar de la pantalla de inicio a la pantalla de aprendizaje.

```
<Button
    android:id="@+id/id_aprender"
    android:layout_width="160dp"
    android:layout_height="65dp"
    android:text="A aprender"
    android:fontFamily="@font/comicsans"
    app:layout_constraintBottom_toBottomOf="@+id/id_practicar"
    app:layout_constraintEnd_toEndOf="@+id/id_practicar"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="@+id/id_practicar"
    app:layout_constraintTop_toTopOf="parent"
    android:onClick="startAprender"
    app:layout_constraintVertical_bias="0.829" />
```

Figura 27. Botón de paso a la pantalla de aprendizaje

```
public void startAprender(View view) {
    startActivity(new Intent( packageContext: this, MainActivity_aprender.class));
}
```

Figura 28. Función relacionada con el botón de la figura 27

4.2.2.2 ImageView

El `ImageView` es un componente para visualizar y manejar el recurso de imagen. Se utiliza este componente para mostrar imágenes y manejar el comportamiento al efectuar un clic sobre la misma imagen. Para la visualización, se utiliza el atributo `android:src` que establece un `drawable` como contenido de la imagen. También tiene la función `android:onClick` mencionada en el apartado 4.2.2.1 para manejar el evento de clic. [28]

```

<ImageView
    android:id="@+id/id_backL"
    android:layout_width="66dp"
    android:layout_height="58dp"
    android:src="@drawable/back_xml"
    android:onClick="back"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

Figura 29. ImageView con el manejo del evento clic

4.2.2.3 TextView

El TextView es un componente de interfaz donde se muestra los textos al usuario, es un componente más utilizado y fundamental en una aplicación de Android. Su atributo android:fontFamily permite utilizar los estilos de texto definidos por el programador. [29]

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="222dp"
    android:layout_weight="1.5"
    android:paddingHorizontal="10dp"
    android:text="    En este modo, tienes tres de vida, cada vez que se
    android:textSize="20dp"
    android:fontFamily="@font/comic_sans"
    android:textColor="@color/lightgreen" />

```

Figura 30. TextView con el estilo definido

4.2.2.4 ViewPager2

El ViewPager2 es una versión mejorada del componente ViewPager, es un contenedor para manejar varios fragmentos y permite al usuario navegar entre pantallas mediante el deslizamiento horizontal o vertical del dedo utilizando su atributo android:orientation. En nuestra aplicación, se utiliza para mostrar las diferentes pantallas de chuleta en sola una pantalla deslizando con el dedo hacia arriba o abajo, o bien haciendo clic en el menú situado a la izquierda de la pantalla. Su implementación la detallaremos en el apartado 4.2.3.5. [30]

```

<androidx.viewpager2.widget.ViewPager2
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:orientation="vertical"
    android:background="@color/lightyellow"
    android:id="@+id/id_viewPager" />

```

Figura 31. ViewPager con orientación vertical

4.2.3 CountdownTimer

CountDownTimer es un temporizador que nos propone Android, con el método onTick() se puede manejar el comportamiento en cada segundo como por ejemplo cambiar un texto. Y el método onFinish() permite decidir qué se hace al finalizar el temporizador. En la nuestra aplicación, este temporizador se utiliza para quitar una de vida cuando se finaliza el temporizador en una partida. [31]

```
//se inicia el cronometro, si pasa el tiempo limite, se resta una de vida
private void initTime(){
    ct = new CountdownTimer( millisInFuture: 16000, countDownInterval: 1000) {
        @Override
        public void onTick(long millisUntilFinished) {
            time.setText("Tiempo: " + millisUntilFinished / 1000);
        }

        @Override
        public void onFinish() {
            restHearth();
            nextExersice();
        }
    }.start();
}
```

Figura 32. El temporizador del modo Ilimitado utilizando CountdownTimer

4.2.4 Service

El Service es un componente de la aplicación cuyo objetivo principal es ejecutar operaciones de larga duración en segundo plano sin necesidad de la interacción con el usuario. Para dicha ejecución, la aplicación debe llamar a la función startService() de la clase Context para pedirle al sistema que programe el trabajo del servicio. [32]

Además, un Service puede proporcionar funcionalidad a otras aplicaciones utilizando la función bindService() de la clase Context, que le permite establecer una conexión persistente con el servicio para poder interactuar con él. [33]

4.2.4.1 Adición de música del fondo

En nuestra aplicación, utilizamos Service para la reproducción de la música de fondo, primero se crea una clase heredado de la clase Service de Android, y con la función onStartCommand (Intent intent, int flags, int startId) para configurar el comportamiento cuando el servicio se comienza.

Como se muestra en la figura 33, primero se declara y se instancia un objeto MediaPlayer y establece la canción que se va a reproducir como la música de fondo en cuanto se crea por primera vez el objeto Service.

```

private MediaPlayer mediaPlayer;

@Override
public void onCreate() {
    try {
        mediaPlayer = new MediaPlayer();
        mediaPlayer = MediaPlayer.create(context: MusicService.this, R.raw.babyspark);
        mediaPlayer.prepare();
    } catch (IllegalStateException | IOException e){
        e.printStackTrace();
    }
    super.onCreate();
}

```

Figura 33. Creación del player cuando se establece el servicio por primera vez

Una vez creado el objeto y cuando se ejecuta el servicio, se empieza a reproducir la música configurando el comportamiento cuando se finaliza o cuando se produce un error.

```

@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    mediaPlayer.start();
    mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
        @Override
        public void onCompletion(MediaPlayer mediaPlayer) {
            try{
                mediaPlayer.start();
            } catch (IllegalStateException e ){
                e.printStackTrace();
            }
        }
    });
    mediaPlayer.setOnErrorListener(new MediaPlayer.OnErrorListener() {
        @Override
        public boolean onError(MediaPlayer mediaPlayer, int i, int i1) {
            try{
                mediaPlayer.release();
            } catch (Exception e){
                e.printStackTrace();
            }
            return false;
        }
    });
    return super.onStartCommand(intent, flags, startId);
}

```

Figura 34. Comienzo de la música de fondo cuando se ejecuta el servicio

Por último, cuando el servicio deja de ejecutar o cuando se destruye, se para la música y se libera el objeto

```

@Override
public void onDestroy() {
    mediaPlayer.stop();
    mediaPlayer.release();
    super.onDestroy();
}

```

Figura 35. Destrucción de Servicio

La música del fondo elegida en la aplicación es el famoso “Baby Shark”, se elige a esta canción porque es una canción simple y repetitiva, y además es una de las más canciones escuchadas (en Youtube aproximadamente 10,5 billones de vistas [34]), y ha sido colado en el puesto 83 del Billboard Global 200 en la semana de 23 de abril del año 2022 [35], es un ranking de canciones más populares del mundo. [36]

4.2.5 Fragment y FragmentStateAdapter

Un fragmento se puede considerar como una parte de la interfaz gráfica de una pantalla. Se puede combinar varios fragmentos en una sola pantalla para crear una vista de múltiples pantallas en una sola o bien utilizar un fragmento para varios actividades por tema de reutilización de código. Cada fragmento tendrá su propio ciclo de vida como las actividades o los servicios, esto evita exceso de códigos en la Activity, por lo tanto, simplifica la lógica de Activity. En la nuestra aplicación, se crea un fragmento por cada pantalla de chuleta que hay, configurando para cada una de ellas su propia chuleta y una figura correspondiente. Para ello, se crea por cada fragmento una clase heredado del Fragment de Android, y un correspondiente Layout donde se configura el texto de chuleta y la visualización de figura. [37]

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".A.fragments.Afragment1">
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text=" Sumar es juntar, unir o agregar varias cosas y agruparlas"
        android:fontFamily="@font/littledaysalt"
        android:textSize="25dp"
        android:paddingTop="30dp"
        android:paddingLeft="10dp"
        android:textColor="@color/black"
        tools:layout_editor_absoluteX="-61dp"
        tools:layout_editor_absoluteY="-43dp" />
    <ImageView
        android:layout_width="351dp"
        android:layout_height="398dp"
        android:src="@drawable/sumaa"
        app:layout_constraintBottom_toBottomOf="@+id/textView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.708" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Figura 36. Layout para el fragmento de la chuleta Suma

```

public class Afragment1 extends Fragment {
    public Afragment1() {
        // Required empty public constructor
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_afragment1, container, attachToRoot: false);
    }
}

```

Figura 37. Fragmento de la chuleta Suma

Una vez creado los fragmentos para Suma, Resta, Multiplicación, División, Triángulo, Cuadrado, Rectángulo, Trapecio y Círculo. Para relacionarlos y combinarlos en una sola pantalla, es necesario el componente ViewPager2 con un Adaptador llamado FragmentStateAdapter. El principal beneficio de este adaptador es cuando se utiliza este adaptador, si un fragmento deja de ser visualizar, se destruirá ese objeto fragmento llamando el método onDestroy() pero guardando el estado ese fragmento, y cuando ese fragmento se vuelve a visualizar, en su método onCreate() se recupera el estado. Esto implica que, si una aplicación tiene muchos fragmentos, no se consumirá muchas memorias. En la nuestra aplicación nos viene bastante bien, porque tenemos nueve fragmentos. [38] [39] [40]

En cuanto a su implementación, primero se crea el objeto ViewPager y relacionarlo con el ViewPager del Layout, después crear un adaptador de páginas pasándole un array de los fragmentos que se desea visualizar en la pantalla, y, por último, asignar este adaptador al ViewPager creado. Para cambiar en la pantalla un fragmento del otro fragmento, se escribe una función para ello y configurarlo en el método registerOnPageChangeCallback de la clase ViewPager.

```

private void initPager() {
    viewPager = findViewById(R.id.id_viewPager);
    ArrayList<Fragment> fragments = new ArrayList<>();
    fragments.add(new Afragment1());
    fragments.add(new Afragment2());
    fragments.add(new Afragment3());
    fragments.add(new Afragment4());
    fragments.add(new Afragment5());
    fragments.add(new Afragment6());
    fragments.add(new Afragment7());
    fragments.add(new Afragment8());
    fragments.add(new Afragment9());
    MyFragmentPagerAdapter pagerAdapter = new MyFragmentPagerAdapter(getSupportFragmentManager(), getLifecycle(), fragments);
    viewPager.setAdapter(pagerAdapter);
    viewPager.registerOnPageChangeCallback(onPageSelected(position) → {
        super.onPageSelected(position);
        changeTab(position);
    });
}

```

Figura 38. Configuración de ViewPager y su Adapter

```

public class MyFragmentPagerAdapter extends FragmentStateAdapter {
    List<Fragment> fragmentList = new ArrayList<>();

    public MyFragmentPagerAdapter(@NonNull FragmentManager fragmentManager, @NonNull Lifecycle lifecycle, List<Fragment> fragments) {
        super(fragmentManager, lifecycle);
        fragmentList = fragments;
    }

    @NonNull
    @Override
    public Fragment createFragment(int position) { return fragmentList.get(position); }

    @Override
    public int getItemCount() { return fragmentList.size(); }
}

```

Figura 39. Clase de Adaptador

4.2.6 Toast

El Toast es un aviso en una ventana pequeña que proporciona información al usuario, Se utiliza este aviso porque no afecta a las actividades que están en curso y las interacciones que está manejando el usuario. Este aviso desaparecerá solo en un tiempo personalizado. En nuestra aplicación se utiliza el Toast para dar un Feedback cuando el jugador está contestando una pregunta, y si se trata de una respuesta correcta, se mostrará frases de elogio en un fondo verde a los niños, y en caso contrario se mostrará frases de ánimo en un fondo rojo. [41] [42]

El estilo del Toast predefinido no es el estilo que se espera para nuestra aplicación, para ello, se crea primero el estilo de la ventana de Toast en una vista. El estilo se guarda en el directorio drawable del proyecto, y la vista en Layout. Para la respuesta correcta, se utiliza el fondo verde y en contrario se utiliza uno rojo.

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- color background -->
    <solid android:color="#BADB66" />
    <!-- margen negro -->
    <stroke
        android:width="1px"
        android:color="#FFFFFF" />
    <!-- cuatro puntas -->
    <corners
        android:bottomLeftRadius="10px"
        android:bottomRightRadius="10px"
        android:topLeftRadius="10px"
        android:topRightRadius="10px" />

    <padding
        android:bottom="5dp"
        android:left="5dp"
        android:right="5dp"
        android:top="5dp" />
</shape>

```

Figura 40. Estilo de la ventana de Toast para la respuesta correcta

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/custom_toast"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/toast_bg"
    android:gravity="center_vertical"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/message"
        android:textSize="20dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp" />

</LinearLayout>

```

Figura 41. Vista de Toast para la respuesta correcta

Una vez creado el estilo, se crea una función para aplicarlo. Para ello, se implementa un método que recibe la frase para mostrar y una variable que indica si es correcta la respuesta seleccionada. En este método, también se define la posición y la duración de Toast sobre la pantalla que se quiere demostrar. Para respuesta correcta y errónea, se ha definido tres frases para cada una de ellas y las almacenamos en dos arrays de String.

```

private void showToast(String str, boolean option){

    LayoutInflater inflater = getLayoutInflater();
    View view;
    TextView message;
    //Se mostrará diferentes vistas en función de la respuesta seleccionada
    if (option) {
        view = inflater.inflate(R.layout.toast_view,
            (ViewGroup) findViewById(R.id.custom_toast));
        message = (TextView) view.findViewById(R.id.message);
    }else {
        view= inflater.inflate(R.layout.toast_view_1,
            (ViewGroup) findViewById(R.id.custom_toast_incorrect));
        message = (TextView) view.findViewById(R.id.message_incorrect);
    }
    message.setText(str);

    Toast toast = new Toast( context: MainActivity_mod01.this);
    toast.setGravity(Gravity.CENTER, xOffset: 0, yOffset: 0);
    toast.setDuration	Toast.LENGTH_SHORT );
    toast.setView(view);
    toast.show();

}

```

Figura 42. Método para demostrar Toast

Por último, se llama a este método en donde se determina si la respuesta es correcta o errónea.

4.2.7 Almacenamiento de datos

El sistema de Android utiliza un sistema de archivos parecido a los sistemas basados en discos, y ofrece varias opciones de almacenamiento como almacenamiento específico de la aplicación, almacenamiento compartido, SharedPreferences que es un almacenamiento donde guarda los datos primitivos y privados, y bases de datos, entre ellas se decide utilizar almacenamiento específico de la aplicación, que es un almacenamiento propio, es decir, cada aplicación tendrá su propio almacenamiento y una de sus características más importantes es la guardar informaciones que otras aplicaciones no deban acceder.

Ese almacenamiento se divide en dos ubicaciones, directorios de almacenamiento interno y directorios de almacenamiento externo. Lo que se diferencia entre ellos es que el almacenamiento interno está encriptado y esto evita el acceso de otras aplicaciones, pero ese tipo de almacenamiento suele ser pequeño por lo que no debe guardarse en allí los archivos grandes, y se suelen guardar informaciones pequeños y sensible como por ejemplo base de datos importantes, SharedPreferences, configuraciones del usuario, etc. En cambio, el almacenamiento externo es un almacenamiento también de uso exclusivo de la propia aplicación, pero dicho almacenamiento puede ser accedido por otras aplicaciones si esas aplicaciones tienen permisos adecuados, se suelen guardar allí las imágenes de alta resolución u otros archivos de gran tamaño.

Ambos tipos tienen ubicaciones que, dependiendo del uso, se divide en la ubicación para almacenar archivos persistentes y la ubicación para almacenar datos en caché. Y cuando el usuario se desinstala la aplicación, se borrará todos los archivos guardados en el almacenamiento específico para esa aplicación, por lo que no debe guardar los archivos interesados por usuario al después de la desinstalación como por ejemplo las fotos, etc.

En la nuestra aplicación, se decide guardar los datos en almacenamiento interno de los archivos persistentes, porque solamente se guardará las seis calificaciones (dos modos * tres niveles de dificultades) que son enteros y resultan ocupar menor tamaño. Para almacenar de esta forma, se utiliza la función `getFilesDir` de la clase `Context`. La clase `Context` es una interfaz global de información proporcionada por el sistema de Android acerca del entorno de la aplicación. Y `getFilesDir` es una función que devuelve el directorio del almacenamiento mencionado para que podamos crear un fichero en ello.

```
public void setTopCal(String s, String filename){
    File file = new File(getFilesDir(),filename);
    try(Writer writer = new FileWriter(file)){
        char []data = s.toCharArray();
        writer.write(data);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Figura 43. Función para guardar el récord en el almacenamiento interno de los archivos persistentes

4.2.8 Adición de colores

En esta sección se explicará la adición de colores. Se ha aplicado dos métodos para añadir los colores a distintos componentes, el primero servirá para añadir colores a los botones mediante configurar el archivo que se viene por defecto al crear un proyecto Android, se llama theme.xml y está situado en el directorio res/values/. En ello se cambia el valor de Ítem colorPrimary para el color de botón y también el valor de Ítem colorOnPrimary para cambiar el color del texto de botón.

```
<resources xmlns:tools="http://schemas.android.com/tools">
  <!-- Base application theme. -->
  <style name="Theme.MathApp" parent="Theme.MaterialComponents.DayNight.DarkActionBar">
    <!-- Primary brand color. -->
    <item name="colorPrimary">@color/lightblue</item>
    <item name="colorPrimaryVariant">@color/purple_700</item>
    <item name="colorOnPrimary">@color/lightgreen</item>
    <!-- Status bar color. -->
    <item name="android:statusBarColor" tools:targetApi="L">?attr/colorPrimaryVariant</item>
    <!-- Customize your theme here. -->
  </style>
</resources>
```

Figura 44. Fichero Theme.xml de la aplicación

Y el otro método serían para el resto de los componentes o para un Layout, como, por ejemplo, el componente TextView, se puede cambiar el color de su texto agregando el atributo android:textColor, o para un Layout el atributo android:background. Este método tiene prioridad sobre el anterior, es decir, si se ha configurado mediante los dos métodos para el color de un botón, se establecerá el color que se configura en este método.

```
<LinearLayout
  android:id="@+id/id_intro1"
  android:layout_width="350dp"
  android:layout_height="300dp"
  android:background="@color/lightblue"
  android:gravity="center"
  android:orientation="vertical"
  app:layout_constraintBottom_toBottomOf="parent"
  app:layout_constraintEnd_toEndOf="parent"
  app:layout_constraintStart_toStartOf="parent"
  app:layout_constraintTop_toTopOf="parent">

  <TextView
    android:layout_width="match_parent"
    android:layout_height="222dp"
    android:layout_weight="1.5"
    android:paddingHorizontal="10dp"
    android:text="En este modo, tienes tres de vida, cada vez que se pase de tiempo"
    android:textSize="20dp"
    android:fontFamily="@font/comicsans"
    android:textColor="@color/lightgreen"/>

  <Button
    android:id="@+id/id_intro1_btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="A_empesap!!"
    android:fontFamily="@font/comicsans"
    android:backgroundTint="@color/azure"
    android:textColor="@color/black"/>

</LinearLayout>
```

Figura 45. El Layout con colores de fondo

4.3.2 Estructura de los elementos de diseño

En cuanto a los elementos de diseño, Android Studio los guarda en el directorio `app/src/main/res`, y en función de tipo de elemento los guardan en diferentes carpetas.

Para los Layouts, se almacena en la carpeta `Layout` de `res`.

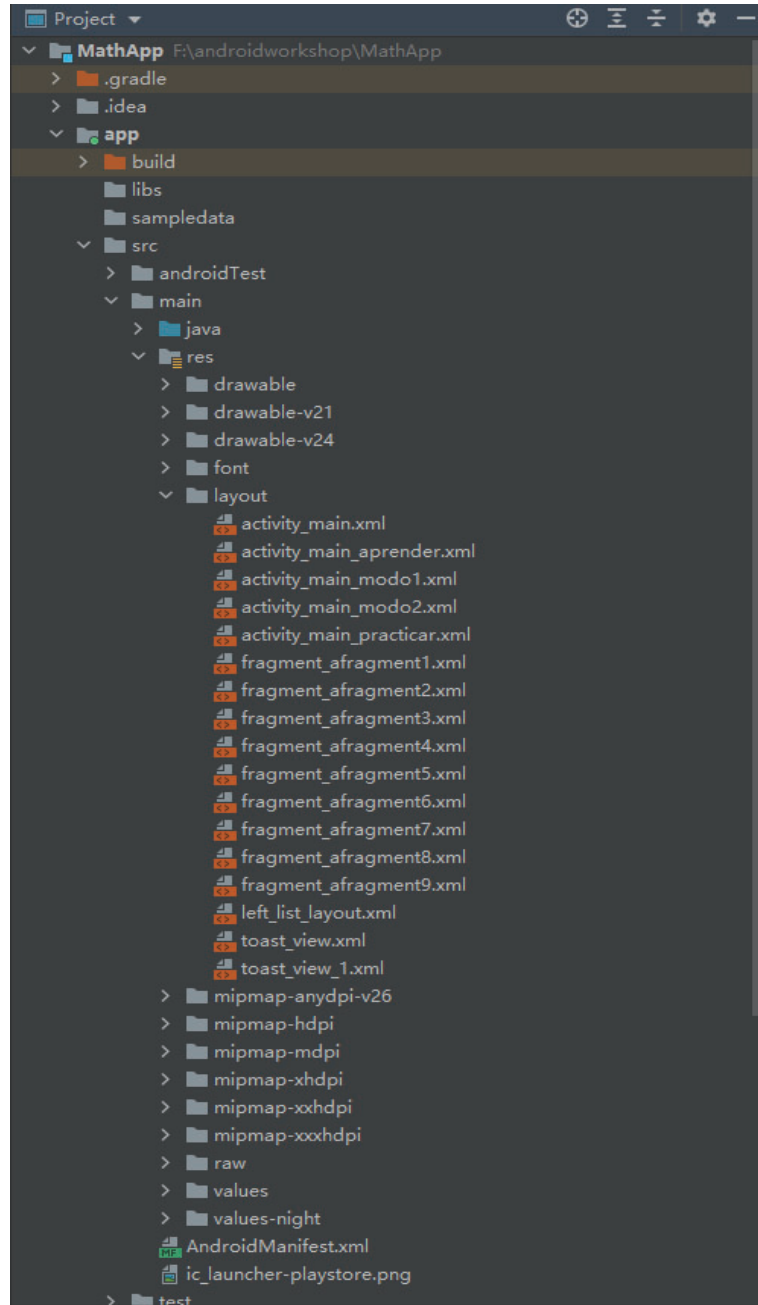


Figura 47. Almacenamiento de Layout

Y para los iconos, las imágenes, el funcionamiento de checkbox se guardan en la carpeta `drawable`. Como se muestra en la figura 48.

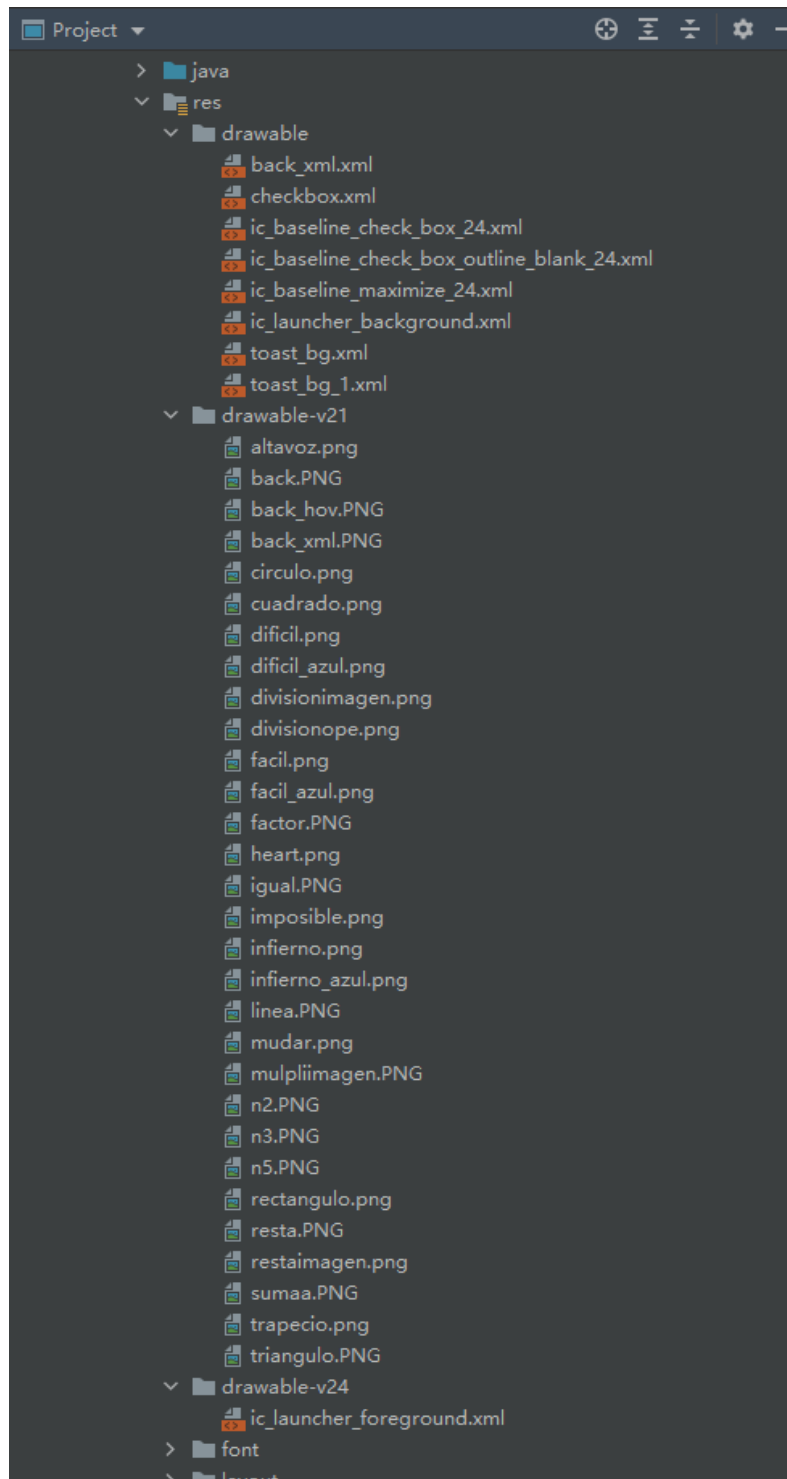


Figura 48. Almacenamiento de drawable

Las fuentes de letras se guardan en la carpeta font, los recursos de música se guardan en la carpeta raw, y por último se guardan en la carpeta values los ficheros donde se define los colores, temas y caracteres de String para la utilización en la aplicación.

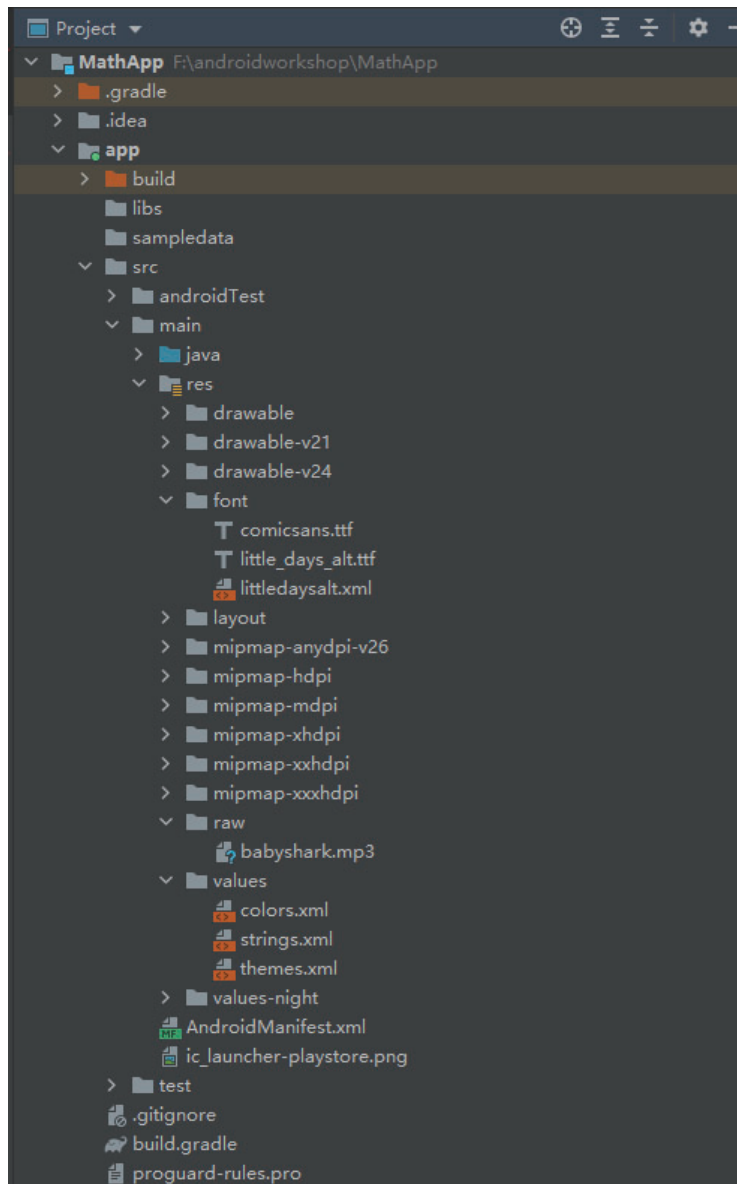


Figura 49. Fuentes, Músicas, restos de elementos

4.4 Pruebas de la aplicación

Una vez terminado el desarrollo y la implementación de las funcionalidades, para asegurar el correcto funcionamiento de la aplicación es necesario realizarse una serie de pruebas para testear las funcionalidades implementadas y también la colaboración de varias funcionalidades.

4.4.1 Pruebas unitarias

En las pruebas unitarias se realiza pruebas para cada funcionalidad implementada. En la tabla 17 se presenta las pruebas realizadas y los resultados obtenidos.

Identificador de prueba	Descripción de Prueba	Comportamiento esperado	Comportamiento obtenido
PU01	Acceder al módulo de aprendizaje	Mostrar la pantalla de aprendizaje con el contenido de Suma	Mostrado la pantalla de aprendizaje con el contenido de Suma
PU02	En la pantalla de aprendizaje, deslizar hacia abajo para cambiar el contenido de Suma a contenido de Resta	Mostrar la pantalla de aprendizaje con el contenido de Resta	Mostrado la pantalla de aprendizaje con el contenido de Resta
PU03	En la pantalla de aprendizaje, haciendo clic para cambiar el contenido de Resta al contenido de Círculo	Mostrar la pantalla de aprendizaje con el contenido de Círculo	Mostrado la pantalla de aprendizaje con el contenido de Círculo
PU04	Volver a la pantalla principal desde la pantalla de aprendizaje	Mostrar la pantalla principal de aplicación	Mostrado la pantalla principal de aplicación
PU05	Empezar una partida de modo Ilimitado con el nivel Fácil	Mostrar la pantalla del modo Ilimitado con la partida empezada	Mostrado la pantalla del modo Ilimitado con la partida empezada
PU06	Contestar una respuesta correcta en el modo Ilimitado.	Sumar 1000 puntos a la calificación final y mostrar la siguiente pregunta con el temporizador restablecido	Sumado 1000 puntos a la calificación final y mostrado la siguiente pregunta con el temporizador restablecido
PU07	Contestar una respuesta errónea en el modo Ilimitado	Quitar una de vida y mostrar la siguiente pregunta con el temporizador restablecido	Quitado una de vida y mostrado la siguiente pregunta con el temporizador restablecido
PU08	Esperar hasta que se acabe el tiempo de una pregunta en el modo Ilimitado	Quitar una de vida y mostrar la siguiente pregunta con el temporizador restablecido	Quitado una de vida y mostrado la siguiente pregunta con el temporizador restablecido
PU09	Empezar una partida de modo Desafío con el nivel Difícil.	Mostrar la pantalla del modo Desafío con la partida empezada	Mostrado la pantalla del modo Desafío con la partida empezada
PU10	Contestar una respuesta correcta en el modo Desafío	Mostrar la siguiente pregunta	Mostrado la siguiente pregunta
PU11	Contestar una respuesta errónea en el modo Desafío	Quitar una de vida y mostrar la siguiente pregunta	Quitado una de vida y mostrado la siguiente pregunta

PU12	Salir en medio de una partida de modo Ilimitado	Mostrar la pantalla de práctica	Mostrado la pantalla de práctica
PU13	Salir en medio de una partida de modo Desafío	Mostrar la pantalla de práctica	Mostrado la pantalla de práctica
PU14	Silenciar la música de fondo	Se deja de reproducir la música de fondo	Se deja de reproducir la música de fondo
PU15	Reproducir la música de fondo	Se vuelve a reproducir la música de fondo	Se vuelve a reproducir la música de fondo

Tabla 17. Pruebas Unitarias

4.4.2 Pruebas del sistema

En las pruebas del sistema se testear la colaboración de varias funcionalidades implementadas. En la tabla 18 se presenta las pruebas realizadas y los resultados obtenidos.

Identificador de prueba	Descripción de Prueba	Comportamiento esperado	Comportamiento obtenido
PS01	Empezar y terminar una partida en modo Ilimitado con nivel fácil, contestando correctamente las primeras 10 preguntas, y después contestar incorrectamente 3 preguntas.	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y el jugador se obtiene 10000 puntos al terminar la partida	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y el jugador se obtiene 10000 puntos al terminar la partida
PS02	Empezar y terminar una partida en modo Ilimitado con nivel Difícil, contestando correctamente las primeras 10 preguntas, y después contestar incorrectamente 3 preguntas.	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y el jugador obtiene 10000 puntos al terminar la partida	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y el jugador obtiene 10000 puntos al terminar la partida
PS03	Empezar y terminar una partida en modo Ilimitado con nivel Infierno, contestando correctamente las primeras 10 preguntas, y después contestar incorrectamente 3 preguntas.	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y el jugador obtiene 10000 puntos al terminar la partida	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y el jugador obtiene 10000 puntos al terminar la partida

PS04	Empezar y terminar una partida en modo Desafío con nivel Fácil, contestando correctamente las 20 preguntas	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y se mostrarán los tiempos consumidos durante la partida	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y se mostrarán los tiempos consumidos durante en la partida
PS05	Empezar y terminar una partida en modo Desafío con nivel Difícil, contestando correctamente las 20 preguntas	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y se mostrarán los tiempos consumidos durante en la partida	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y se mostrarán los tiempos consumidos durante en la partida
PS06	Empezar y terminar una partida en modo Desafío con nivel Infierno, contestando correctamente las 20 preguntas	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y se mostrarán los tiempos consumidos durante en la partida	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y se mostrarán los tiempos consumidos durante en la partida
PS07	Empezar y terminar una partida en modo Desafío con nivel Fácil, contestando incorrectamente las 3 preguntas	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y saltará un mensaje indicando que ha fracasado el jugador	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y saltará un mensaje indicando que ha fracasado el jugador
PS08	Empezar y terminar una partida en modo Desafío con nivel Difícil, contestando incorrectamente las 3 preguntas	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y saltará un mensaje indicando que ha fracasado el jugador	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y saltará un mensaje indicando que ha fracasado el jugador
PS09	Empezar y terminar una partida en modo Desafío con nivel Infierno, contestando incorrectamente las 3 preguntas	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y saltará un mensaje indicando que ha fracasado el jugador	Los operadores aritméticos y el rango de números de operación corresponden a los diseñados, y saltará un mensaje indicando que ha fracasado el jugador

Tabla 18. Pruebas del sistema

5 Resultados y Evaluación

En esta sección se presentará el resultado final de la aplicación mostrando todas las pantallas finales diseñadas y implementadas. Y por último se realizan unas pruebas de accesibilidad a los niños de un colegio sobre la aplicación desarrollada.

5.1 Resultado de la aplicación

A continuación, se muestran todas las pantallas implementadas y diseñadas de la aplicación final divididas en diferentes apartados según las pantallas.

5.1.1 Captura de la pantalla de inicio

En este apartado se muestra la pantalla que se visualiza al entrar en la aplicación. Dispone dos botones para acceder a los dos módulos diferentes y otro botón para silenciar o reproducir la música del fondo.

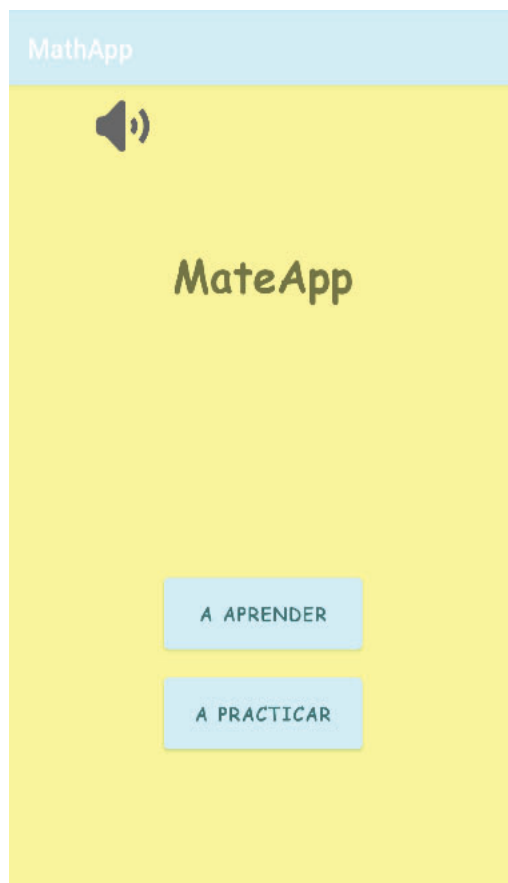


Figura 50. Pantalla de inicio

5.1.2 Captura de la pantalla del módulo práctica

En la pantalla del módulo de práctica, se dispone una parte para que el usuario seleccione el nivel de dificultad y dos botones para empezar la partida de distintos modos, y encima de dichos botones se muestran la mejor puntuación histórica que obtuvo el jugador.

En la parte superior-izquierda, están los botones para volver a la pantalla principal o silenciar la música del fondo.

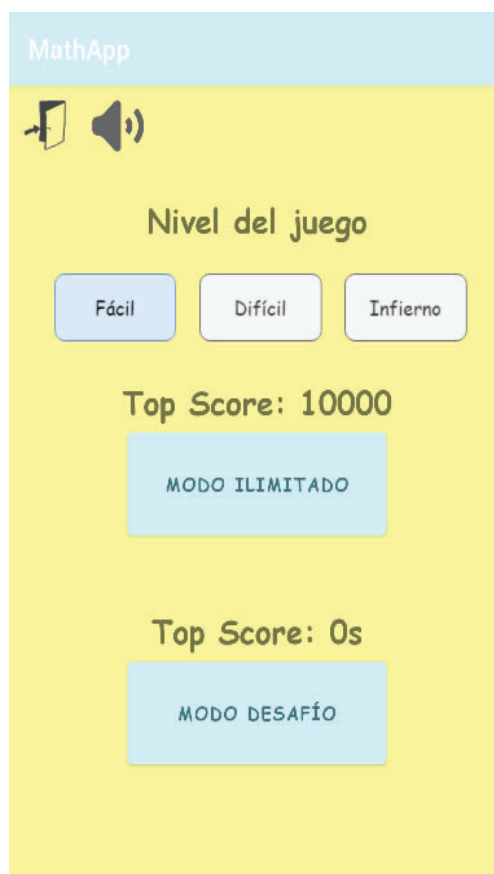


Figura 51. Pantalla del módulo práctica

5.1.3 Capturas de la pantalla de modo ilimitado

Para presentar el modo Ilimitado, se concluye en tres pantallas. Como se muestra en la figura 52, la primera pantalla consiste en una introducción de este modo y al dar al botón “A EMPEZAR”, se inicia la partida. También aporta algunas informaciones más al jugador como las vidas que tendrá él para una partida de juego.

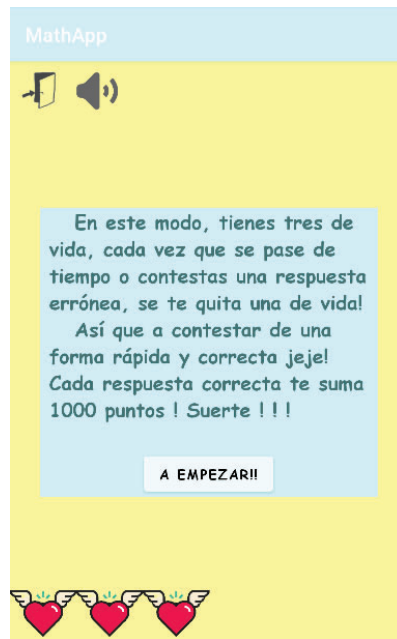


Figura 52. Introducción de modo Ilimitado

Una vez iniciada la partida, el jugador podrá visualizarse la pregunta y las cuatro respuestas para elegir, también podrá conocer en la parte superior-derecha la puntuación que lleva durante la partida, las vidas que le quedan y el tiempo restante para la pregunta actual. Como se muestra en la figura 53, al contestar una pregunta dando un botón de la respuesta, se mostrará diferentes Feedback al jugador según si la haya contestado correcta o incorrectamente.

Cuando el jugador no le queda ninguna vida, se le informará de la puntuación final obtenida y al dar este botón, se vuelve a la pantalla del módulo práctica.

En cualquier momento, se dispone siempre dos botones, uno para salir de la partida, y el otro para silenciar la música del fondo.

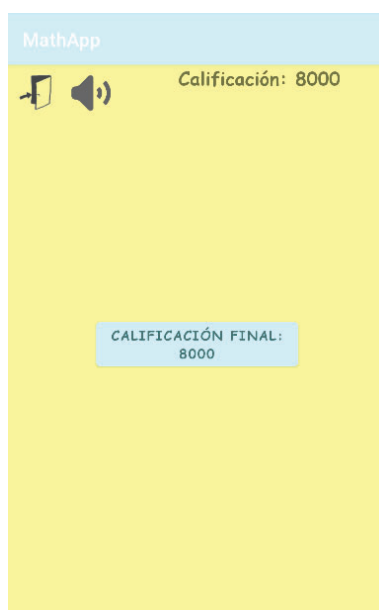


Figura 54. Juego de Modo Ilimitado

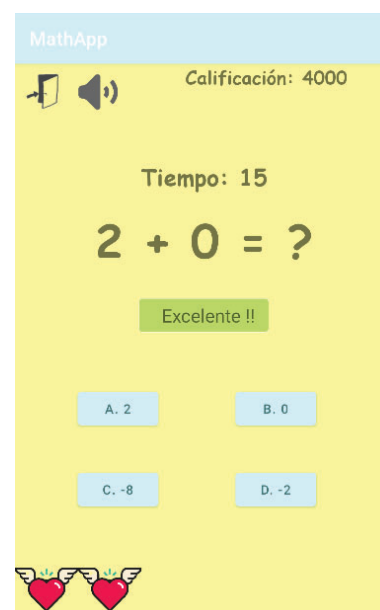


Figura 53. Finalización de Modo Ilimitado

5.1.4 Captura de la pantalla de modo Desafío

A continuación, se muestran las pantallas del modo Desafío, la visualización de este modo es parecida al anterior, la diferencia principal está en el proceso de la partida. Como se muestra en la figura 56, a parte de las vidas de jugador, la pregunta y las respuestas, se visualiza también las preguntas restantes para finalizar la partida, y en la parte superior-derecha se muestra el contador de tiempo.

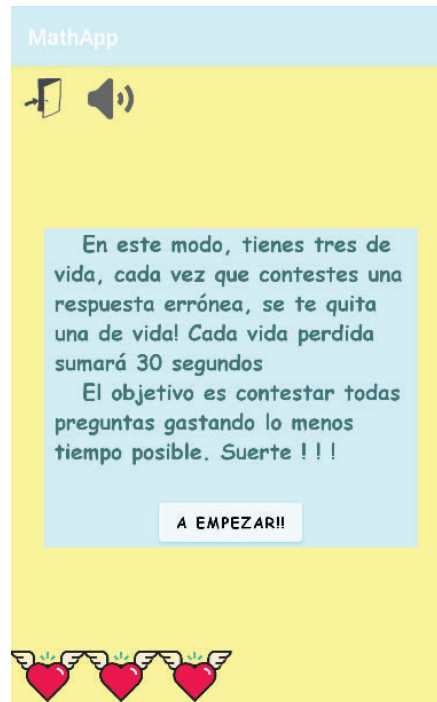


Figura 55. Introducción de modo Desafío

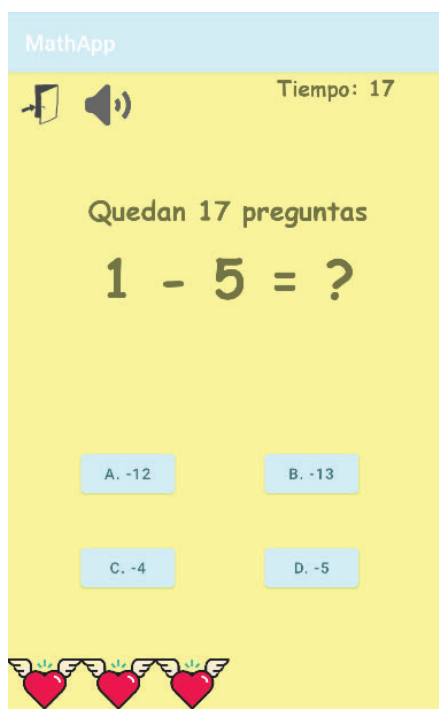


Figura 56. Juego de Modo Desafío

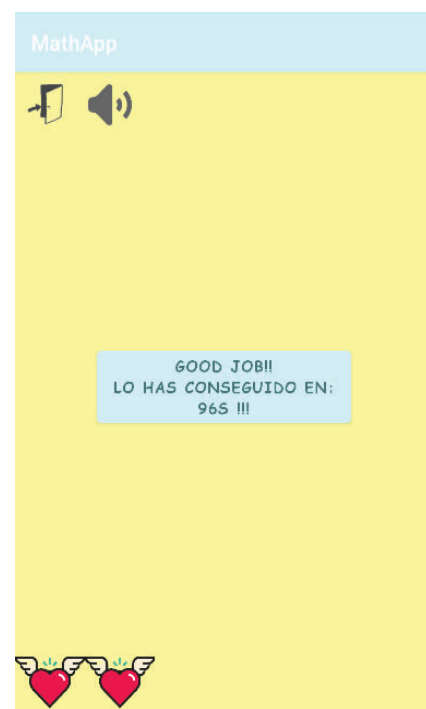


Figura 57. Finalización de Modo Desafío

5.1.5 Capturas de la pantalla del módulo aprendizaje

Por último, se presenta el módulo de aprendizaje, para ello, están nueve pantallas cada una presentando el contenido de suma, resta, multiplicación, división, triángulo, cuadrado, rectángulo, trapecio y círculo. Se puede cambiar el contenido deslizando hacia arriba o abajo o bien haciendo clic en el contenido que le interese al usuario. En todas estas pantallas siempre se dispone un botón para volver a la pantalla principal y otro botón para silenciar/reproducir la música del fondo.



Figura 58. El contenido de Suma

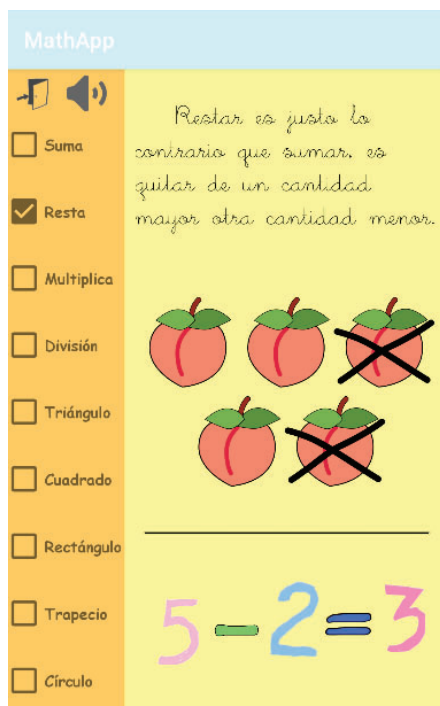


Figura 59. El contenido de Resta

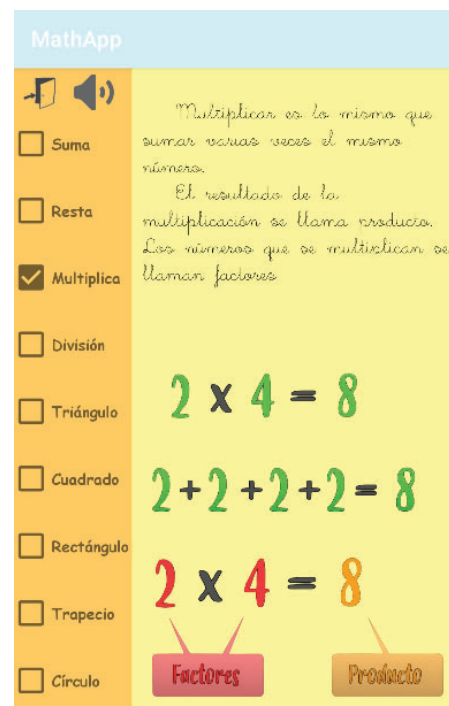
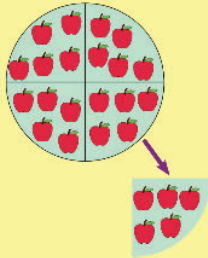


Figura 60. El contenido de Multiplicación

MathApp

Suma
 Resta
 Multiplica
 División
 Triángulo
 Cuadrado
 Rectángulo
 Trapecio
 Círculo

Dividir es la operación más generosa porque significa repartir!
 Dividir es coger una cantidad de algo y repartirla en partes igual.



$$20 \div 4 = 5$$

Figura 61. El contenido de División

MathApp

Suma
 Resta
 Multiplica
 División
 Triángulo
 Cuadrado
 Rectángulo
 Trapecio
 Círculo

Se llama triángulo, trigono o trigonode al polígono de tres lados. Los puntos comunes a cada par de lados se denominan vértices del triángulo.
 Área: $(\text{Base} \times \text{altura}) / 2$
 Perímetro: $\text{ladoA} + \text{ladoB} + \text{ladoC}$




Figura 62. El contenido de Triángulo

MathApp

Suma
 Resta
 Multiplica
 División
 Triángulo
 Cuadrado
 Rectángulo
 Trapecio
 Círculo

Un cuadrado en geometría es un cuadrilátero regular, es una figura plana de cuatro lados congruentes y paralelos dos a dos, y cuatro ángulos interiores rectos (90°).
 Área: $\text{lado} \times \text{lado}$
 Perímetro: $\text{lado} \times 4$


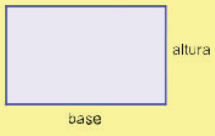


Figura 63. El contenido de Cuadrado

MathApp

Suma
 Resta
 Multiplica
 División
 Triángulo
 Cuadrado
 Rectángulo
 Trapecio
 Círculo

Un rectángulo es una figura geométrica que posee cuatro ángulos interiores de 90°. Es un paralelogramo, es decir, todos sus lados son paralelos dos a dos.
 Área: $\text{base} \times \text{altura}$
 Perímetro: $\text{base} \times 2 + \text{altura} \times 2$



$A = \text{Base} \times \text{Altura}$

Figura 64. El contenido de Rectángulo

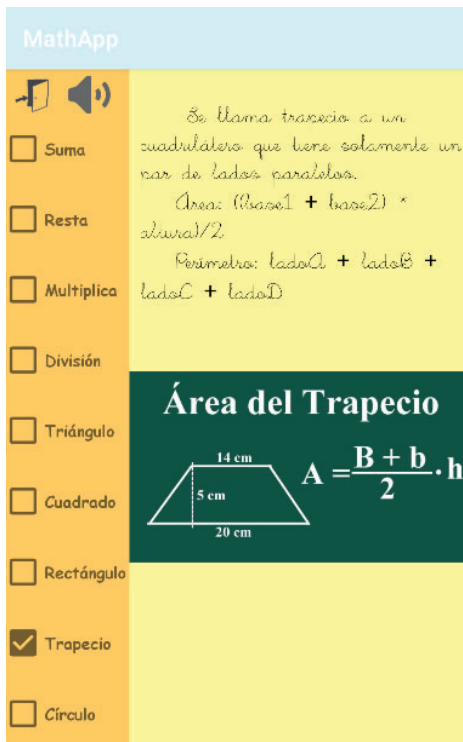


Figura 65. El contenido de Trapecio

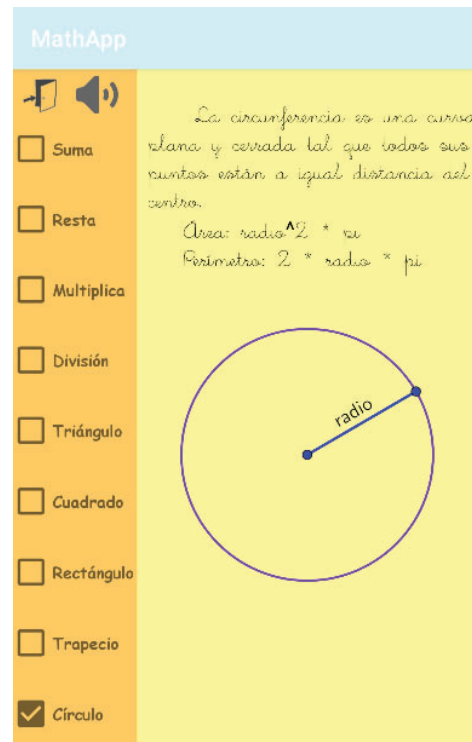


Figura 66. El contenido de Círculo

5.2 Análisis de usuario

En esta sección se realizan varias pruebas de accesibilidad a los usuarios principales de esta aplicación, es decir los niños de primaria. Para ello, se ha ido a un colegio público cerca para hacer las pruebas a los niños de distintos cursos, este colegio se llama “Colegio Público Juan Sebastián Elcano”. El procedimiento de la prueba consiste en primero informar al niño una tarea específico por ejemplo acceder al contenido de Trapecio, y después apuntar los datos importantes durante la realización de tarea. Para hacer dichas pruebas, siempre se ha solicitado permiso de los tutores de los niños que realizan las pruebas.

Los datos de pruebas se presentarán en forma de tablas mostrando los datos más importantes.

5.2.1 Prueba de accesibilidad 1

Información que recibieron los niños de colegio: Intenta encontrar una chuleta de la operación Resta.

Los pasos óptimos para dicha prueba:

1. Pulsar el botón “A APRENDER”.
2. Haciendo clic a la sección Resta o bien deslizar hacia abajo.

Curso	Tarea finalizada	Número de pasos	Forma de acceder a Resta(Clic/Deslizamiento)	Tiempo
Primero	Sí	13	Clic	53”
Primero	Sí	4	Clic	25”
Segundo	Sí	7	Clic	33”
Segundo	Sí	4	Clic	21”
Segundo	Sí	10	Clic	43”
Tercero	Sí	2	Clic	20”
Cuarto	Sí	4	Clic	23”
Cuarto	Sí	2	Clic	17”
Quinto	Sí	2	Clic	18”
Sexto	Sí	2	Clic	10”

Tabla 19. Resultado de la primera prueba de accesibilidad

5.2.2 Prueba de accesibilidad 2

Información que recibieron los niños de colegio: Empezar una partida del modo Ilimitado con el nivel de juego Difícil, contestar una pregunta y volver a la pantalla principal.

Los pasos óptimos para dicha prueba:

1. Pulsar el botón “A PRACTICAR”.
2. Seleccionar el nivel Difícil
3. Pulsar el botón “MODO ILIMITADO”.
4. Pulsar el botón “A EMPEZAR”
5. Seleccionar una de las cuatro respuestas
6. Salir de la partida pulsando el icono de puerta
7. Volver a la pantalla principal pulsando el icono de puerta

Curso	Tarea finalizada	Número de pasos	Tiempo
Primero	No, se le ha olvidado seleccionar el nivel difícil	10	1’06”
Primero	Sí, pero no ha utilizado icono para salir de la partida	9	1’18”
Segundo	Sí	9	1’07”
Tercero	Sí	9	59”
Tercero	No, se le ha olvidado seleccionar el nivel difícil	11	1’12”
Cuarto	Sí	10	1’03”
Quinto	Sí	7	38”

Quinto	No, se le ha olvidado seleccionar el nivel difícil	6	43''
Sexto	Sí	7	42''
Sexto	Sí	7	41''

Tabla 20. Resultado de la segunda prueba de accesibilidad

5.3 Problemas encontrados

En esta sección se mostrarán los problemas encontrados durante la realización de las pruebas de accesibilidad de los usuarios, se analizará cuáles son los defectos que provocan dichos problemas, y se intentará sacar una solución para suprimir esos defectos y evitar dichos problemas, hace que la aplicación sea más intuitiva y accesible.

Como se muestra en la primera prueba de accesibilidad, se puede ver que los usuarios para cambiar de un contenido al otro contenido lo hacen siempre por medio de clic, ya que es difícil saber a primera hora sin ninguna indicación previa hacer el cambio por medio de deslizamiento. También por medio de clic es más intuitivo y rápido en algunos casos, dado que, si se desea pasar el primer contenido al último, los usuarios sólo tienen que hacer un clic en vez de ocho deslizamientos. Por lo tanto, parece que el cambio por deslizamiento es una funcionalidad excesiva. También se detecta que la mitad de los niños a la primera vez que se han metido al módulo de prácticas para encontrar la chuleta de Resta, se cree que es porque no está muy claro y clasificado en la pantalla principal los botones de entrada a los dos módulos, sus diferencias son muy pocas solamente en el texto, para mejorar eso, se puede cambiar los botones de texto por los botones de imagen ya que las imágenes son más intuitivas y se diferencian mejor para los niños.

De la segunda prueba de accesibilidad, se intenta buscar la intuición en el módulo de práctica, y se detecta tres problemas principales. El primer problema ya está mencionado anteriormente que consisten en los botones de entrada a los dos módulos de aplicación. El segundo problema consiste en que los usuarios se les olvidan seleccionar el nivel de juego, se cree que es porque los colores de botones son pálidos, para mejorarlo se puede hacerlos en colores más atractivos incluso con imágenes.

6 Conclusiones y Líneas futuras

Para terminar, se presentará la conclusión de este proyecto y las posibles mejoras futuras, también se contará las dificultades que se ha encontrado durante la realización del proyecto y la implementación de la aplicación final.

6.1 Conclusiones

Durante el desarrollo del proyecto, se han adquirido los conocimientos básicos con el desarrollo de una aplicación Android desde cero, conocer qué es Android, en qué consiste su entorno de desarrollo, etc. También es muy importante, para el futuro profesional esta experiencia de programar los dispositivos móviles con uno de los sistemas operativos más comunes.

Durante el desarrollo del proyecto, se han encontrado distintos problemas tanto en la implementación de aplicación como en el desarrollo de TFG, y gracias a mi tutor y mis amigos, podía seguir avanzando en el proyecto superando las dificultades surgidas a lo largo del desarrollo. Sin duda, sus ayudas forman una parte indispensable del proyecto, sin ellos no podría terminar este trabajo de fin de grado. El mayor problema encontrado ha sido principalmente en aprender y familiarizarme con el desarrollo de Android, porque se trata de una disciplina totalmente desconocida para el autor.

En definitiva, ha sido un completo placer dedicarnos a un proyecto con una tecnología que no se había visto nunca, y me siento muy satisfecho de ampliar mis conocimientos en el área del desarrollo de aplicaciones móviles. Los objetivos planteados en las fases iniciales han sido superados, y este trabajo de fin de grado supone un gran aprendizaje en mi desarrollo profesional.

6.2 Definición de líneas futuras

Aparte del análisis de problema y las mejoras propuestas en el apartado 5.3 Problemas encontrados, entre las otras mejoras de la aplicación, estaría muy bien implementar también la aplicación en IOS, así soportando a los dos principales sistemas operativos para las familias que solamente tienen los dispositivos IOS. También será muy interesante la adición de los módulos de otras disciplinas de la educación primaria, completando las asignaturas obligatorias hace que esta aplicación tenga más utilidad, y que pueda desarrollar las habilidades de los niños en múltiple áreas. Todavía quedan muchas mejoras por hacer, por ejemplo, agregar más músicas del fondo incluso permitiendo la subida de música personal; la adición de una funcionalidad de usuarios para que los niños puedan ver sus calificaciones óptimas en distintos dispositivos móviles, etc. Estas son algunas mejoras que se pueden agregar en funcionalidades futuras.

7 Análisis de Impacto

En este capítulo se analizará el impacto potencial de la aplicación obtenida durante la realización del TFG haciendo referencia a los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030. [43]



Figura 67. Los Objetivos de Desarrollo Sostenible

La aplicación obtenida está dirigida al público infantil y está muy claro la relación entre la aplicación con la educación de calidad, el objetivo de aplicación no es simplemente proporcionar una herramienta para que los niños puedan desarrollar su habilidad matemáticas, sino también para que se estudien con alegría y despertar su interés. Por otro lado, también se puede vincular con la industria, la innovación y la infraestructura, ya que queremos innovar la industria de enseñanzas infantiles.



Figura 68. Objetivo de Desarrollo Sostenible 4: Educación de Calidad

8 Bibliografía

- [1] J. J. García, «Las asignaturas de Educación Primaria,» [En línea]. Available: <https://www.mundoprimeria.com/pedagogia-infantil-primaria/las-asignaturas-de-educacion-primaria>.
- [2] «¿Cuál es la materia favorita de los niños?,» 2022. [En línea]. Available: <https://todosloshechos.es/cual-es-la-materia-favorita-de-los-ninos>.
- [3] S. R. -. Pontevedra, «"Las matemáticas deberían ser la asignatura favorita de los niños",» 2012. [En línea]. Available: <https://www.farodevigo.es/pontevedra/2012/01/17/matematicas-deberian-asignatura-favorita-ninos-17693779.html>.
- [4] Ecoaula, «¿Para qué sirven las matemáticas? Aportan 5 beneficios para la mente,» 2021. [En línea]. Available: <https://www.economista.es/ecoaula/noticias/11095424/03/21/Para-que-sirven-las-matematicas-Aportan-5-beneficios-para-la-mente.html>.
- [5] «Conoce la importancia de la educación primaria en tus hijos,» [En línea]. Available: <https://blog.cambridgecali.edu.co/conoce-importancia-educacion-primaria-en-tus-hijos>.
- [6] «Estado del arte de TFG: qué es y ejemplos,» 2021. [En línea]. Available: <https://tutfgamedida.com/tfg-tfm/estado-del-arte-de-tfg-que-es-y-ejemplos/>.
- [7] «Introducción a Android Studio,» 2021. [En línea]. Available: <https://developer.android.com/studio/intro>.
- [8] «Cómo ejecutar apps en Android Emulator,» 2022. [En línea]. Available: <https://developer.android.com/studio/run/emulator?hl=es-419>.
- [9] «Introducción a Java,» 2021. [En línea]. Available: <https://www.manualweb.net/java/introduccion-java/>.
- [10] «Cómo analizar datos XML,» 2021. [En línea]. Available: <https://developer.android.com/training/basics/network-ops/xml?hl=es-419>.
- [11] ANDROFAST, «XML en Android: conceptos básicos y diferentes tipos de archivos XML utilizados,» 2017. [En línea]. Available: <https://www.androfast.com/2017/11/xml-en-android-conceptos-basicos-y.html>.
- [12] VictorD3D, «¿Qué es Diagrams.net?,» 2020. [En línea]. Available: <https://conocimientolibre.mx/que-es-diagrams-net/>.
- [13] «¿Qué es Google Drive?,» 2017. [En línea]. Available: <https://biblio.fcm.unc.edu.ar/2017/08/02/que-es-google-drive-para-que-nos-sirve/>.
- [14] «Font Awesome ¿Qué es y cómo se usa?,» 2014. [En línea]. Available: <https://www.aquihaydominios.com/blog/font-awesom-que-es-y-como-se-usa/>.

- [15] V. González, «Color favorito según tu generación [Infografía],» 2017. [En línea]. Available: <https://www.occ.com.mx/blog/color-favorito-segun-tu-generacion/>.
- [16] «Colores, formas y sonidos que sorprenden a los niños,» 2016. [En línea]. Available: <https://www.misenal.tv/noticias/para-grandes/colores-formas-y-sonidos-que-sorprenden-a-los-ninos>.
- [17] E. Esteban, «Cómo influyen los colores en la conducta y emociones de los niños,» 2021. [En línea]. Available: <https://www.guiainfantil.com/blog/educacion/conducta/como-influyen-los-colores-en-la-conducta-y-emociones-de-los-ninos/>.
- [18] F. C. Vega, «¿Qué tipos de letra para niños son mejores cuando aprenden a leer?,» [En línea]. Available: <https://www.smartick.es/blog/lectura/tipos-de-letra-para-ninos/>.
- [19] J. Cruz, «¿Con qué tipo de letra hacemos leer a los niños de primero de Primaria?,» 2013. [En línea]. Available: <https://clbe.wordpress.com/2013/07/30/con-que-tipo-de-letra-hacemos-leer-a-los-ninos-de-primero-de-primaria/>.
- [20] «Ciclo de vida de desarrollo de Software,» [En línea]. Available: <https://www.efectodigital.online/single-post/2018/04/23/ciclo-de-vida-de-desarrollo-de-software>.
- [21] «Modelo en Cascada,» 2021. [En línea]. Available: https://es.ryte.com/wiki/Modelo_en_Cascada.
- [22] ORACLE, «Descripción de flujos de navegación,» [En línea]. Available: https://docs.oracle.com/cloud/help/es/pbcs_common/PFUSA/understanding_navigation_flows_100xdb2e13be.htm#PFUSA-GUID-E5A0FF0F-ECE5-4D00-8CE5-2710D364C8FE.
- [23] Lucidchart, «Qué es un diagrama de flujo,» 2022. [En línea]. Available: <https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-flujo>.
- [24] «LinearLayout,» 2022. [En línea]. Available: <https://developer.android.com/reference/android/widget/LinearLayout>.
- [25] «Cómo crear una IU responsiva con ConstraintLayout,» 2022. [En línea]. Available: <https://developer.android.com/training/constraint-layout?hl=es-419>.
- [26] «Botones,» 2021. [En línea]. Available: <https://developer.android.com/guide/topics/ui/controls/button>.
- [27] «R.attr,» 2022. [En línea]. Available: <https://developer.android.com/reference/android/R.attr#onClick>.
- [28] «ImageView,» 2022. [En línea]. Available: <https://developer.android.com/reference/android/widget/ImageView>.
- [29] «TextView,» 2022. [En línea]. Available: <https://developer.android.com/reference/android/widget/TextView>.
- [30] «Cómo crear vistas deslizantes con pestañas mediante ViewPager2,» 2020. [En línea]. Available:

<https://developer.android.com/guide/navigation/navigation-swipe-view-2?hl=es-419>.

- [31] «CountDownTimer,» 2022. [En línea]. Available: <https://developer.android.com/reference/android/os/CountDownTimer>.
- [32] «Service,» 2022. [En línea]. Available: <https://developer.android.com/reference/android/app/Service>.
- [33] «Descripción general de los servicios,» 2021. [En línea]. Available: <https://developer.android.com/guide/components/services?hl=es-419>.
- [34] P. B. S. -. K. S. & Stories, «Baby Shark Dance | #babys shark Most Viewed Video | Animal Songs | PINKFONG Songs for Children,» 2016. [En línea]. Available: https://www.youtube.com/watch?v=XqZsoesa55w&ab_channel=PinkfongBabyShark-Kids%27Songs%26Stories.
- [35] «Billboard Global 200,» 2022. [En línea]. Available: <https://www.billboard.com/charts/billboard-global-200/>.
- [36] J. BOTE, «'Baby Shark' Has Crashed Into the Pop Charts. How Did It Get There?,» 2019. [En línea]. Available: <https://www.npr.org/2019/01/09/683651912/baby-shark-has-crashed-into-the-pop-charts-how-did-it-get-there?t=1650550455307>.
- [37] «Fragmentos,» 2021. [En línea]. Available: <https://developer.android.com/guide/components/fragments?hl=es-419>.
- [38] «FragmentManager,» 2022. [En línea]. Available: <https://developer.android.com/reference/androidx/viewpager2/adapter/FragmentManager>.
- [39] «FragmentManagerAdapter,» 2022. [En línea]. Available: <https://developer.android.com/reference/androidx/fragment/app/FragmentManagerAdapter>.
- [40] AlexMomotov, «Diferencia entre fragmentManager y fragmentstatepageradapter,» 2013. [En línea]. Available: <https://www.iteramos.com/pregunta/21834/diferencia-entre-fragmentpageradapter-y-fragmentstatepageradapter>.
- [41] «Toast,» 2022. [En línea]. Available: <https://developer.android.com/reference/android/widget/Toast>.
- [42] «Descripción general de los avisos,» 2021. [En línea]. Available: <https://developer.android.com/guide/topics/ui/notifiers/toasts>.
- [43] «Objetivos de Desarrollo Sostenibles,» [En línea]. Available: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>.

9 Anexos

9.1 Índice de figuras


Figura 1. El logo de Android Studio	3
Figura 2. Administrador de dispositivos virtuales de Android.....	4
Figura 3. El logo de Java	5
Figura 4. XML.....	5
Figura 5. El logo de diagrams.net	6
Figura 6. El logo de Google Drive	7
Figura 7. El logo de Font Awesome	7
Figura 8. Ejemplo de la galería de iconos FontAwesome.....	8
Figura 9. Las actitudes que transmiten los colores a los niños [17]	10
Figura 10. Las letras de tipografía Comic Sans MS	10
Figura 11. Las letras de tipografía Escolar	11
Figura 12. Ciclo de vida del software.....	11
Figura 13. Ciclo de vida cascada.....	12
Figura 14. Ejemplo de un flujo de navegación	17
Figura 15. El flujo de la navegación de pantalla	18
Figura 16. Ejemplos de diagramas de flujo.....	19
Figura 17. Diagrama de flujo del sistema de juego.....	20
Figura 18. Boceto de la pantalla principal.....	21
Figura 19. Boceto de la pantalla de aprendizaje.....	21
Figura 20. Boceto de la pantalla de práctica.....	22
Figura 21. Boceto de la pantalla de introducción del modo de juego.....	23
Figura 22. Boceto de la pantalla de modo Ilimitado	24
Figura 23. Boceto de la pantalla de modo Desafío.....	24
Figura 24. Boceto de la pantalla de resultado.....	25
Figura 25. Ejemplo de LinearLayout	26
Figura 26. Ejemplo de ConstraintLayout	26
Figura 27. Botón de paso a la pantalla de aprendizaje	27
Figura 28. Función relacionada con el botón de la figura 27	27
Figura 29. ImageView con el manejo del evento clic.....	28
Figura 30. TextView con el estilo definido.....	28
Figura 31. ViewPager con orientación vertical	28
Figura 32. El temporizador del modo Ilimitado utilizando CountdownTimer. 29	
Figura 33. Creación del player cuando se establece el servicio por primera vez	30
Figura 34. Comienzo de la música de fondo cuando se ejecuta el servicio	30
Figura 35. Destrucción de Servicio	30
Figura 36. Layout para el fragmento de la chuleta Suma	31
Figura 37. Fragmento de la chuleta Suma	32
Figura 38. Configuración de ViewPager y su Adapter	32
Figura 39. Clase de Adaptador	33
Figura 40. Estilo de la ventana de Toast para la respuesta correcta	33
Figura 41. Vista de Toast para la respuesta correcta	34
Figura 42. Método para demostrar Toast.....	34
Figura 43. Función para guardar el récord en el almacenamiento interno de los archivos persistentes	35
Figura 44. Fichero Theme.xml de la aplicación.....	36

Figura 45. El Layout con colores de fondo.....	36
Figura 46. Estructura de código en Android Studio	37
Figura 47. Almacenamiento de Layout	38
Figura 48. Almacenamiento de drawable.....	39
Figura 49. Fuentes, Músicas, restos de elementos	40
Figura 50. Pantalla de inicio.....	44
Figura 51. Pantalla del módulo práctica.....	45
Figura 52. Introducción de modo Ilimitado.....	46
Figura 53. Finalización de Modo Ilimitado.....	46
Figura 54. Juego de Modo Ilimitado.....	46
Figura 55. Introducción de modo Desafío.....	47
Figura 56. Juego de Modo Desafío.....	47
Figura 57. Finalización de Modo Desafío	47
Figura 58. El contenido de Suma.....	48
Figura 59. El contenido de Resta.....	48
Figura 60. El contenido de Multiplicación	48
Figura 61. El contenido de División.....	49
Figura 62. El contenido de Triángulo	49
Figura 63. El contenido de Cuadrado.....	49
Figura 64. El contenido de Rectángulo.....	49
Figura 65. El contenido de Trapecio.....	50
Figura 66. El contenido de Círculo.....	50
Figura 67. Los Objetivos de Desarrollo Sostenible	54
Figura 68. Objetivo de Desarrollo Sostenible 4: Educación de Calidad.....	54

9.2 Índice de tablas

Tabla 1. R1. Seleccionar el módulo de aplicación.....	13
Tabla 2. R2. Cambiar de una chuleta a la otra mediante clic	13
Tabla 3. R3. Cambiar de una chuleta a la otra mediante deslizamiento	14
Tabla 4. Silenciar o reproducir la música de fondo	14
Tabla 5. R5. Seleccionar el nivel de dificultad.....	14
Tabla 6. R6. Seleccionar el modo de juego.....	14
Tabla 7. R7. Mostrar el récord de cada modo y en cada nivel	14
Tabla 8. R8. Iniciar una partida.....	15
Tabla 9. R9. Mostrar la introducción de cada modo.....	15
Tabla 10. R10. Seleccionar la respuesta de una pregunta	15
Tabla 11. R11. Mostrar los puntos que lleva durante una partida.....	15
Tabla 12. R12. Mostrar el tiempo restante para la pregunta.....	15
Tabla 13. R13. Mostrar las preguntas restantes	16
Tabla 14. R14. Mostrar las vidas restantes.....	16
Tabla 15. R1. Legibilidad.....	16
Tabla 16. R2. Usabilidad	16
Tabla 17. Pruebas Unitarias	42
Tabla 18. Pruebas del sistema	43
Tabla 19. Resultado de la primera prueba de accesibilidad	51
Tabla 20. Resultado de la segunda prueba de accesibilidad	52

Este documento esta firmado por



Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
Fecha/Hora	Sun May 29 20:35:08 CEST 2022
Emisor del Certificado	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
Numero de Serie	561
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)