



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
Mención en Ingeniería del Software

SmartSpend
Una aplicación móvil educativa para introducir
a los niños en la gestión del dinero

Alumno: Álvaro de la Rosa Zarzuelo
Tutor: Juan A. Muñoz Cristobal

A mi padre, que estaría orgulloso de este trabajo.

Agradecimientos

A mi familia, por siempre apoyarme en cada paso de mi camino.

A mis amigos, por ayudarme en los momentos difíciles.

A mis compañeros, que me han acompañado en esta etapa de crecimiento y aprendizaje.

A mi tutor Juan, que me ha ayudado a poner rumbo a este trabajo y sin cuya ayuda nada de esto habría sido posible.

Resumen

Este documento describe el proceso de diseño y desarrollo de una aplicación móvil educativa diseñada para introducir a los niños, de forma dinámica y entretenida, en las habilidades básicas para la gestión del dinero, presupuestar y ahorrar. Se emplean técnicas de simulación y una interfaz que trata de conseguir un acercamiento a las actuales aplicaciones bancarias, todo ello junto a pequeños juegos para crear una experiencia educativa y entretenida. La tecnología empleada es un desarrollo nativo en Android, empleando Kotlin, y un *back-end* remoto con FireBase.

Abstract

This document describes the design and development process of an educational mobile application designed to introduce children, in a dynamic and entertaining way, to basic money management skills, budgeting, and saving. Simulation techniques have been used along with an interface that aims to mimic current banking applications, all combined with small games to create an educational and entertaining experience. The technology used is native Android development using Kotlin, and a remote FireBase *back-end*.

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Motivación	2
1.3. Objetivos	3
1.3.1. Objetivos del proyecto	3
1.3.2. Objetivos personales	3
1.4. Estructura de la memoria	3
2. Educación financiera en aplicaciones móviles	5
2.1. Estudio sobre la educación financiera	5
2.1.1. Edad idónea para introducir las finanzas en la educación	5
2.1.2. Estrategias para enseñar sobre finanzas	6
2.2. Aplicaciones parecidas	7
3. Planificación y metodología	11
3.1. Metodología iterativa e incremental junto a <i>kanban</i>	11
3.2. ¿Qué es <i>Kanban</i> ?	11
3.3. ¿Qué es el desarrollo iterativo e incremental?	12
3.4. Aplicación de las metodologías al proyecto	13
3.4.1. ¿Por qué estas metodologías?	13

3.4.2. Aplicación al proyecto	14
3.5. Planificación inicial	15
3.6. Plan de riesgos	16
3.6.1. Gestión inicial de riesgos	17
3.6.2. Gestión inicial de oportunidades	23
3.7. Estimación de costes	24
3.8. Monitorización de los riesgos	26
3.9. Comparación respecto a la planificación inicial	27
4. Tecnologías	29
4.1. Herramientas de comunicación	29
4.2. Herramientas para el análisis, diseño y documentación	29
4.3. Herramientas para la gestión y control del proyecto	30
4.4. Herramientas para el desarrollo del proyecto	31
5. Análisis	33
5.1. Requisitos	33
5.1.1. Requisitos funcionales	33
5.1.2. Requisitos no funcionales	34
5.2. Casos de uso	35
5.3. Modelo de dominio	37
5.4. Diagramas de secuencia	39
6. Diseño	43
6.1. Diseño de la interfaz de usuario	43
6.1.1. Investigación de un diseño apropiado	43
6.1.2. Guías de diseño seguidas	44
6.2. Prototipo inicial de la interfaz	45

6.3. Prueba de satisfacción del usuario	51
6.3.1. Planificación de la prueba	51
6.3.2. Desarrollo de la prueba	52
6.4. Arquitectura de la aplicación	55
6.4.1. ¿Por qué un desarrollo nativo en Android?	57
6.5. Back-end	57
6.5.1. Estructura de la base de datos	59
6.5.2. Reglas de seguridad	62
6.6. Front-end	63
6.6.1. Patrones de diseño	63
6.6.2. Estructura de la aplicación	64
7. Implementación y pruebas	67
7.1. Formación previa	67
7.2. Iteraciones de desarrollo	67
7.2.1. Primera iteración (12/3/2024 - 9/4/2024)	67
7.2.2. Segunda iteración (9/4/2024 - 22/4/2024)	68
7.2.3. Tercera iteración (22/04/2024 - 06/05/2024)	69
7.2.4. Cuarta iteración (06/05/2024 - 14/05/2024)	70
7.2.5. Quinta iteración (14/05/2024 - 23/05/2024)	71
7.2.6. Sexta iteración (23/05/2024 - 30/05/2024)	72
7.3. Organización del código	72
7.4. Dificultades y retos	73
7.5. Política de privacidad	74
7.6. Seguridad de la aplicación	74
7.7. Licencia	74
7.8. Pruebas y evaluación de la app	75
7.8.1. Casos de pruebas	75

8. Conclusiones	89
8.1. Conclusiones	89
8.2. Planes de futuro	90
8.3. Valoración personal	91
Bibliografía	93
A. Manual de instalación	97
B. Manual de usuario	99
B.1. Inicio de sesión y registro en la aplicación	99
B.2. Establecer contraseña de perfil	100
B.3. Crear nuevos miembros de la familia	102
B.4. Crear una asignación	102
B.5. Gestión de las tareas	103
B.6. Añadir gastos	104
B.7. Consultar movimientos de dinero de los menores	105
B.8. Crear y alcanzar objetivos de ahorro	106
C. Política de privacidad	109
D. Enlaces adicionales	115

Capítulo 1

Introducción

1.1. Contexto

En el último informe PISA sobre las competencias financieras (2018), el rendimiento de España, de 492 puntos, quedó significativamente por debajo del promedio de los países que pertenecen a la OCDE (Organización para la Cooperación y el Desarrollo Económicos), que se encuentra en los 505 puntos. Además solo el 62,8% del alumnado fue capaz de aplicar esos conocimientos financieros en situaciones reales y relevantes para ellos [1], [2].

En la actualidad existen algunos estudios que investigan cómo puede afectar la educación financiera temprana en el comportamiento de ahorro cuando ya son adultos. En la mayoría se llega a la conclusión de que una adecuada enseñanza sobre el ahorro y las finanzas cuando son niños afecta de forma positiva, incrementando la probabilidad de que la persona tenga una mejor responsabilidad financiera cuando sea un adulto [3]-[5].

Además, tras la pandemia, se ha evidenciado un notable aumento en el uso de dispositivos móviles, sobre todo en los más jóvenes. Esto se debe a una combinación de factores como la reducción de los precios de estos dispositivos, un aumento de la conectividad y la constante aparición de contenido para los niños y adolescentes [6]. A raíz de esto han nacido corrientes en contra de que los niños usen el móvil y se está estudiando su prohibición en colegios e institutos. Pero a pesar de esta discusión existente sobre la idoneidad del uso de teléfonos en las aulas [7], su uso sigue muy extendido en el día a día fuera de estas.

Como no se pueden eliminar los dispositivos móviles por completo de la vida de los más jóvenes, otra opción es aprovecharlo para su beneficio, ya que al final no es más que una herramienta y todo depende de cómo se utilice [8]. Por eso, mediante el uso de las tecnologías móviles, se puede intentar favorecer el aprendizaje de conocimientos en materia de finanzas en edades tempranas.

1.2. Motivación

El uso generalizado de dispositivos móviles es evidente en la vida cotidiana, con personas constantemente pegadas a las pantallas, ya sea en el transporte público, durante descansos del trabajo o incluso tomando un café [9]. Las motivaciones también son dispares, puede ser para contestar un mensaje, jugar a un juego, leer un libro o incluso aprender. Además esto último se ha visto potenciado en los últimos años debido al incremento de las aplicaciones que tienen como objetivo el autoaprendizaje, por ejemplo algunas de las más populares son Duolingo [10], para el aprendizaje de nuevos idiomas, u otras otras como Udemy[11], para impartir cursos a distancia.

Estos cambios sociales al final han terminado llegando también a los más pequeños, incentivado también por los meses de confinamiento por Covid-19, en los cuales se emplearon estas tecnologías para mantener entretenidos a los niños [12]. Por esto cada vez es más común ver a niños con un teléfono móvil o tablet en sus manos, normalmente prestado de sus padres, aunque existen casos menos comunes en el que disponen del suyo propio. Incluso algunos padres tienden a recompensar a sus hijos con pasar algo de tiempo con estos dispositivos, normalmente en juegos [13].

Dado que resulta, en cierto modo, inevitable que los niños pasen tiempo frente a las pantallas y, además, lo hagan en aplicaciones poco educativas y quizás incluso adictivas, ¿por qué no aprovechar esta situación para que aprendan de una manera mas innovadora e interactiva? De igual manera, se puede aplicar este enfoque a una habilidad fundamental en el día a día de una persona adulta y que a menudo se le presta poca atención en las aulas: la administración del dinero. De ahí nace mi idea de crear una aplicación móvil, orientada para familias, que funcione como una herramienta para enseñar a los niños estos conceptos.

Es bastante común encontrarse con personas adultas que se enfrentan a dificultades para llegar a fin de mes y, que en muchos casos, la raíz de este problema no radica tanto en la cantidad de ingresos, sino más bien en una mala gestión del dinero. Una mala planificación de los gastos del mes y futuros gastos que afrontar, así como las compras compulsivas sin pensar en las consecuencias, hacen que una parte de la población pase por problemas económicos injustificados [5]. Una forma de solucionar este problema es enseñar buenos hábitos desde una edad temprana. Se ha demostrado que las personas a las que se les enseñan desde niños el valor del dinero y cómo administrarlo de la forma correcta tienen mejores hábitos ahorradores de adultos. Sin embargo, darles dinero únicamente no funciona como estrategia de enseñanza, lo mejor es combinarlo con practicas en la toma de decisiones, objetivos de ahorro y un seguimiento de las finanzas [4].

Este proyecto tratará de dar a las familias una herramienta para enseñar a los niños dentro del entorno familiar. Para ello, previo a la elaboración de la aplicación, se realizará un estudio de cómo orientarla, con el objetivo de que sea lo más completa y óptima posible.

1.3. Objetivos

1.3.1. Objetivos del proyecto

El objetivo principal es desarrollar una aplicación móvil educativa diseñada para enseñar a los niños las habilidades fundamentales de administración financiera, incluyendo la gestión del dinero y el ahorro, todo ello de una manera dinámica y amena. Este objetivo principal se divide en los siguientes objetivos secundarios:

- La aplicación debe permitir a los padres asignar a sus hijos una propina.
- La aplicación debe permitir a los niños crear objetivos de ahorro.
- La aplicación debe enseñar conceptos básicos de economía de forma dinámica y entretenida.
- La aplicación debe iniciar a los niños en el proceso de gestión de gastos e ingresos.

1.3.2. Objetivos personales

Al contar este proyecto también con un carácter académico tiene como objetivo que el alumno adquiera nuevos conocimientos profesionales. Estos objetivos son principalmente:

- Aprender a dirigir y planificar un proyecto desde cero, afrontando los problemas y retos que surjan.
- Mejorar las competencias del alumno en el desarrollo de aplicaciones móviles.
- Aprender Kotlin, debido a que poco a poco está sustituyendo a Java como lenguaje de desarrollo de aplicaciones móviles en Android.
- Mejorar las habilidades del alumno en el diseño de interfaces y diseño de software.

1.4. Estructura de la memoria

Tras esta introducción de la motivación del proyecto y sus objetivos, el resto de la memoria se estructura de la siguiente manera:

Capítulo 2. Estudio de campo. Estudio previo a la realización del Trabajo de Fin de Grado (TFG) que incluye metodologías de enseñanza y estudio de mercado.

Capítulo 3. Planificación. Planificación del proyecto, definición de los riesgos a afrontar, definición del marco del trabajo y como se aplicará a este proyecto.

Capítulo 4. Tecnologías. Presentación y breve explicación de las tecnologías y aplicaciones empleadas para la realización del proyecto.

Capítulo 5. Análisis. Definición de los requisitos del proyecto y decisiones de diseño.

Capítulo 6. Diseño. Diseño de la aplicación tanto a nivel de interfaz como a nivel de arquitectura.

Capítulo 7. Implementación y pruebas. Seguimiento del desarrollo de la aplicación junto con las pruebas de aceptación.

Capítulo 8. Conclusiones. Conclusiones a las que se ha llegado tras completar el proyecto y propuestas de cara al futuro.

Apéndice A. Manual de instalación. Explicación sobre como instalar la aplicación en un dispositivo Android.

Apéndice B. Manual de usuario. Proporciona instrucciones de cómo utilizar la aplicación.

Apéndice C. Política de privacidad.

Apéndice D. Enlaces adicionales.

Capítulo 2

Educación financiera en aplicaciones móviles

2.1. Estudio sobre la educación financiera

En esta sección, se lleva a cabo un estudio sobre como orientar la aplicación de manera apropiada. Inicialmente, se investiga la franja de edad a la que debe dirigirse la aplicación y cual es su forma de aprendizaje. Posteriormente, se exploran las estrategias más efectivas para transmitir los conceptos sobre finanzas.

2.1.1. Edad idónea para introducir las finanzas en la educación

Diversos estudios sobre educación coinciden en que el mejor rango de edad para introducir a los niños en la enseñanza sobre economía se encuentra entre los 7 y 12 años (durante el periodo de educación primaria)[14]. Es durante este periodo cuando los niños ya son capaces de comprender ideas más complejas y aplicarlas a situaciones reales, lo que habilita la comprensión de los conceptos económicos más básicos.

Según el libro *Economics for Kids: Ideas for Teaching in the Elementary Grades* [15], aunque hay variación entre cada niño, se han hallado coincidencias sobre como piensan acerca de conceptos económicos. Se han encontrado evidencia de que el razonamiento económico se desarrolla de manera escalonada, de manera similar a las etapas de pensamiento cognitivo de *Piaget* [16], siendo dichas etapas:

- **1ª etapa:** sin comprensión del sistema. Los niños hasta los siete años piensan que las compras son solo rituales. Muchos creen, por ejemplo, que los bienes simplemente llegan a la tienda, sin que haya un intercambio involucrado.

- **2ª etapa:** dos sistemas no conectados. Los niños en este nivel comprenden que hay que pagar por los bienes y servicios que se ofrecen. Sin embargo, no siempre comprenden que el dinero que emplean las tiendas para traer más bienes, producirlos u ofrecer un servicio, proviene previamente de los clientes.
- **3ª etapa:** dos sistemas integrados. A partir de los diez años, aproximadamente, se desarrolla una mejor comprensión de la relación entre los precios de compra y venta.

Aunque esta investigación no identifica un patrón de pensamiento para cada concepto económico, esta secuencia de desarrollo se ha corroborado para conceptos como *precio*, *dinero*, *banca* e *intercambio*. La clave para una buena enseñanza es poder reconocer los patrones de desarrollo que existen en el pensamiento de los niños y luego diseñar actividades acordes a ello. El económico, al igual que otros tipos de pensamiento, se desarrolla lentamente y mejora con la edad, la labor de las lecciones es asegurar y mejorar el desarrollo del mismo.

2.1.2. Estrategias para enseñar sobre finanzas

Se han estudiado diversas fuentes sobre cuáles son las mejores metodologías para iniciar a los niños en la gestión del dinero [17]-[21]. En algunos puntos, como enseñar los conceptos fundamentales o marcar metas de ahorro, coinciden la mayoría de las fuentes consultadas, y otras veces cada fuente aporta su propia propuesta. Se han seleccionado las estrategias más viables para su implementación en una aplicación móvil.

Antes de comenzar a enumerar las estrategias, se deben mencionar los cuatro conceptos clave de la educación financiera que debemos intentar transmitir: gastos, ingresos, ahorro e inversión [20], [22], aunque este último concepto puede ser demasiado complejo para nuestra franja de edad, por lo que queda descartado.

Las estrategias recopiladas para integrar en la aplicación incluyen:

- Enseñar los conceptos claves: abordar de manera clara y comprensible los conceptos de gastos, ingresos y ahorro.
- Otorgar una cantidad de dinero de manera periódica: permitir que los niños administren una cantidad de dinero de forma regular, fomentando la toma de decisiones financieras.
- Definir objetivos y metas de ahorro: la definición de unas metas puede promover el ahorro para alcanzarlas.
- Llevar un registro detallado: incentivar el registro de los ingresos y gastos, destacando cada depósito o retiro. Esto asentará una serie de buenos hábitos que les será de gran ayuda cuando lleguen a edades adultas.
- Enseñar cálculos sencillos: incentivar el desarrollo y práctica de las habilidades matemáticas, sobre todo la capacidad de cálculo, que son la base de la contabilidad.
- Planificar compras: listar las necesidades y saber diferenciar entre necesidades básicas y caprichos, puede evitar las compras impulsivas.

- Enseñar el valor del dinero: mostrar que el dinero no viene de la nada, la gente debe trabajar para ganarlo.

Respecto a la asignación de dinero que los padres deben hacer a sus hijos, se deben considerar varios puntos importantes [19], [23]:

- No existe un único sistema de asignación que funcione con todas las familias.
- Cualquier sistema puede ser válido, cada familia deberá encontrar el que mejor se adapte a su forma de vida.
- Muchos padres pueden ver como un derroche algunas de las acciones de sus hijos y, por ello dejar de darles dinero de forma periódica, sin embargo, los errores forman parte de su aprendizaje.
- La asignación ayuda que tanto los niños como los padres tengan un mayor control sobre las finanzas, no se trata únicamente de dinero extra o un premio.
- Evitar «acudir al rescate» de los niños. Estos deben de ver las consecuencias de todas sus decisiones, tanto malas como buenas y aprender de ello.

Estas estrategias de aprendizaje no se pueden enseñar únicamente de manera conceptual, deben de ponerse en práctica para que sean efectivas en desarrollar buenos hábitos. La aplicación debe intentar abordar todos estos puntos de la forma más completa posible y, además, intentar hacerlo de forma entretenida y amena para fomentar la continuidad de su uso.

2.2. Aplicaciones parecidas

Un buen análisis de mercado es crucial para el éxito de un proyecto. Debemos conocer el entorno en el que va a existir la aplicación, la competencia y nuestra audiencia. Esto nos proporcionará información valiosa para la toma de decisiones y el diseño, además de determinar su rentabilidad y la viabilidad de entrar a competir.

Un factor a destacar dentro de este mercado es que la mayoría de las aplicaciones son incompatibles con versiones de Android actuales. Esto, seguramente, se debe a que los desarrolladores han dejado de dar soporte porque no generaban beneficios. Este fracaso se puede deber a múltiples factores, como malos sistemas de monetización, una inadecuada campaña de marketing o la falta de ella, la pérdida del interés del público o desinterés inicial, un mal diseño y/o la incapacidad de hacer competencia al resto de opciones del mercado.

Aun así existen potenciales competidores con el mismo objetivo que nosotros, o con objetivos diferentes pero que se aproximan de forma indirecta. Se ha realizado una recopilación de estas aplicaciones y se han analizado al detalle para que nuestra aplicación marque una diferencia respecto a las opciones existentes.

De todas las aplicaciones encontradas merece la pena destacar las siguientes:

- **FamZoo** [24]: aplicación para gestionar el dinero que gastan tus hijos, combinando los conceptos de las tarjetas prepago y el tradicional método de la «propina». Tiene funcionalidades interesantes como la de administrar el presupuesto familiar o gráficos que muestran el movimiento del dinero. El mayor problema de esta aplicación es que su interfaz es incómoda, sobrecargada y en ocasiones confusa, más cercana a unas hojas de cálculo, lo que la hace prácticamente imposible de manejar para unos niños.
- **Mi dinero y Yo** [25]: aplicación que fundamentalmente consiste en lecciones mediante texto acompañado con dibujos hechos para niños. El problema es que las lecciones se pueden hacer largas y tediosas, lo que puede hacer que los niños pierdan la atención e interés rápidamente.
- **Money Kids** [26]: un juego por niveles bastante sencillo en el que tienes que comprar un producto y te dan una serie de monedas para llegar al pago. Un punto a destacar de la aplicación es que tiene compatibilidad con varios tipos de moneda del mundo. Sin embargo, la aplicación tiene un gran problema y es que todos los productos que se muestran son productos reales y al acabar el nivel se te redirige a una aplicación de compras, lo cual es peligroso si los padres son descuidados y dejan la tarjeta puesta para comprar.
- **The Game of Life 2** [27]: adaptación del juego de mesa con el mismo nombre. En este juego se simula la vida a través de sus diferentes etapas, desde la juventud hasta la jubilación, avanzando por casillas. Al final gana el que haya conseguido más metas o dinero a lo largo de su vida. Este juego puede ayudar a los más pequeños a tener una aproximación a la gestión del dinero, sin embargo, se centra demasiado en ser un juego y menos en el aprendizaje. Además como barrera de entrada el juego no es gratis.
- **Monopoly** [28]: adaptación a dispositivos móviles del juego de mesa mundialmente conocido. Al estar también basado en un juego de mesa tiene las mismas virtudes y defectos que *The Game of Life 2*, siendo este también de pago.
- **Bankaroo** [29]: es un banco virtual diseñado para niños donde pueden aprender el valor del dinero de forma práctica. Les permite familiarizarse con conceptos financieros, como presupuestar, ahorrar y tomar decisiones. Es una de esas aplicaciones con poco soporte que se mencionan previamente, dificultando incluso simplemente probarla.
- **NatWest Rooster Money** [30]: es una aplicación y tarjeta de prepago diseñada para niños. Permite a los padres establecer propinas y enseñar a los niños sobre la gestión financiera y el ahorro para metas específicas. Sin embargo, únicamente esta disponible en Reino Unido.

A modo de resumen se ha diseñado una tabla que facilita el análisis de las principales características de las aplicaciones:

CAPÍTULO 2. EDUCACIÓN FINANCIERA EN APLICACIONES MÓVILES

Nombre	FamZoo	Mi dinero y Yo	Money Kids	The Game of Life 2	Monopoly	Bankaroo	NatWest Rooster Money
Involucra participación de la familia	Sí	No	No	Puede	Puede	Sí	Sí
Interfaz adecuada	No	Si	Si	Si	Si	Si	No
Administración de dinero	Si	No	No	Si	Si	Si	Si
Objetivos de ahorro	Si	No	No	No	No	Si	Si
Lecciones educativas	No	Si	No	No	No	No	No
Elementos de juego	No	Si	Si	Si	Si	No	No
Disponibilidad	Completa	Completa	Completa	Completa	Completa	Poco soporte	Solo UK
Precio	Gratuita (con versión de pago)	Gratuita	Gratuita (con publicidad agresiva)	4,99€	4,99€	Gratuita (con versión de pago)	Gratuita (con versión de pago)
Audiencia a la que va dirigida	Padres e hijos	Niños	Niños	Todo el mundo	Todo el mundo	Padres e hijos	Padres e hijos

Figura 2.1: Análisis de aplicaciones rivales en el mercado

En resumen, nos encontramos frente a un mercado poco explotado, que generalmente pasa desapercibido, facilitando una entrada. A pesar de ello, es crucial idear estrategias para destacar entre las opciones existentes. Con este proyecto se intentará conseguir un producto mejor y más completo, con una interfaz más adecuada y características mejoradas, buscando sobresalir en el mercado. También reconocemos la importancia de una campaña de marketing efectiva, pero dado que este proyecto es de naturaleza académica, esta parte queda fuera del alcance del trabajo de fin de grado.

Capítulo 3

Planificación y metodología

3.1. Metodología iterativa e incremental junto a *kanban*

Para la planificación y seguimiento de este proyecto se ha optado por emplear una combinación de un desarrollo iterativo e incremental con metodología Kanban. Estas dos metodologías por separado tienen filosofías y objetivos diferentes, juntos se pueden complementar para conseguir un desarrollo más eficiente y flexible. Esto se detallará más adelante en la sección 3.4.

3.2. ¿Qué es *Kanban*?

Kanban [31] es un método para gestionar el trabajo y mejorar la eficiencia del desarrollo de productos y/o servicios. Se basa en hacer visible el trabajo, con el objetivo de asegurar que la carga es la adecuada. Para realizar esto, se emplea un sistema de flujo de tareas que limita la cantidad de trabajo en progreso, empleando para ello tarjetas visuales. Este mecanismo de señalización se muestra en tableros *Kanban* que ayudan a representar los límites del trabajo en progreso. Esta representación se hace mediante columnas que representan las diferentes etapas del desarrollo (ver figura 3.1).

Kanban limita la cantidad de trabajo mediante un WIP (Work In Progress) limit. De esta forma se evita sobrecargar al equipo, se mantiene un trabajo constante y se identifican los cuellos de botella. Se emplea un sistema mediante el cual el trabajo es “trasladado” a través del tablero, pasando de una columna a otra en función de la etapa en la que se encuentre, y no se da inicio a nuevas tareas hasta que otras son completadas o bloqueadas, lo que deja capacidad disponible.

En resumen, *Kanban* es una metodología basada en volver más transparentes las tareas para todo el equipo, facilitando la comunicación y colaboración entre los miembros, todo ello

gracias a que todos tienen más información sobre el estado del proyecto. Es reconocido por su simplicidad y flexibilidad, lo que permite adoptarlo en diversos proyectos. Más concretamente, en el desarrollo de software, Kanban se utiliza en combinación con otros enfoques para mejorar la gestión y la eficiencia del flujo de trabajo.

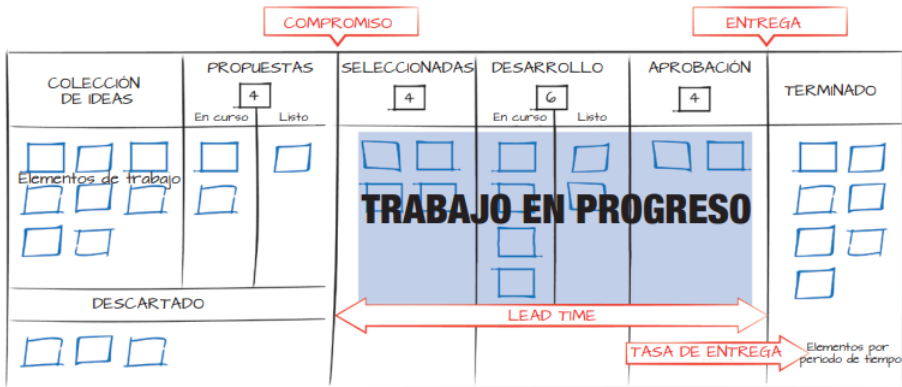


Figura 3.1: Ejemplo de un tablero Kanban [31]

3.3. ¿Qué es el desarrollo iterativo e incremental?

El desarrollo iterativo e incremental [32], [33] es un método de planificación basado en el desarrollo y mejora de un producto de forma progresiva mediante iteraciones a lo largo del proyecto. El objetivo es reducir la complejidad y la incertidumbre de un desarrollo de software, permitiendo la adaptación a medida que se cambian y ajustan los requisitos.

Se trata de un proceso dividido en ciclos en los que se marcan unos objetivos de mejora. Cada uno de los ciclos se divide en fases: planificación, diseño, implementación y pruebas. Al final de la iteración se produce un ejecutable del que se recopila una retroalimentación que será empleada para realizar un ajuste del alcance, corregir fallos y mejorar el producto (ver figura 3.2). Estos ciclos también son incrementales, es decir, el desarrollo se realiza de forma gradual, agregando nuevas funcionalidades o mejorando las existentes. Para mantener un control del proyecto se van realizando entregas parciales, para proporcionar una visión más tangible del progreso.

Este método de trabajo es conocido sobre todo por su flexibilidad, siendo adecuado para proyectos donde los requisitos no están completamente definidos al comienzo, donde el alcance puede variar durante el desarrollo y que necesitan de pruebas continuas para obtener retroalimentación temprana. Permite a los desarrolladores adaptarse a medida que se tiene más información y se comprenden mejor los requisitos del proyecto.



Figura 3.2: Fases del desarrollo iterativo e incremental

3.4. Aplicación de las metodologías al proyecto

3.4.1. ¿Por qué estas metodologías?

El desarrollo iterativo e incremental es una metodología ágil, pero ¿por qué decantarse por esta elección frente a otras metodologías ágiles?, existen una serie de razones para ello:

- Los requisitos del proyecto no están completamente definidos desde el comienzo y puede haber cambios. La metodología iterativa e incremental nos permite realizar los ajustes necesarios.
- Al ser un trabajo que lleva bastante tiempo, en comparación con otros proyectos académicos, necesitamos cierta flexibilidad y capacidad para manejarlo, esta metodología nos lo ofrece dividiendo el proyecto en partes más pequeñas y asumibles que se reparten en iteraciones.
- Se tiende a producir entregas más frecuentes y tempranas, lo que permite obtener retroalimentación y demostrar progreso regularmente.
- La metodología iterativa e incremental tiene menos ceremonias y roles que otras alternativas. Por ejemplo, Scrum tiene una serie de roles como el scrum master, product owner, etc. Para la realización del TFG, donde colaboran únicamente dos personas, se prefiere una estructura menos formal sin tantos roles y evitando las reuniones definidas, que se sustituyen por otras más flexibles a lo largo del proyecto.

- Un mayor control sobre el proceso de desarrollo, permitiendo adaptarse a las necesidades que surjan. Tiene ciclos de retroalimentación más frecuentes, lo que permite una mejora continua del producto.

Por otro lado, Kanban se ha añadido como una segunda metodología de apoyo debido a que provee una manera de mejorar la gestión de tareas y la eficiencia del flujo de trabajo. Facilita la comunicación y sincronización entre tutor y alumno, dando además una visión más transparente del trabajo realizado y el avance del proyecto.

3.4.2. Aplicación al proyecto

Las dos metodologías previamente mencionadas nos proporcionan un marco ágil y flexible para gestionar las tareas y adaptarse a los cambios, cuya aplicación puede resumirse en una serie de puntos clave y pasos a seguir en el desarrollo.

- Comenzar identificando los requisitos iniciales, aunque estos puedan cambiar, eliminarse o añadirse nuevos en el futuro, nos ayudará a establecer un alcance inicial. También se debe crear el tablero Kanban correspondiente para visualizar las tareas y el flujo del trabajo, proporcionando una visión más clara del estado del proyecto.

- En el tablero existirán cinco columnas: lista de tareas, en proceso, completadas, en revisión y cerradas (ver figura 3.3). A continuación se describe el flujo de trabajo con el tablero.

- Las tareas pasarán de *lista de tareas* a *en proceso* cuando el alumno esté trabajando en ellas y una vez completadas pasan a *completadas*.
- Cuando el tutor revise los resultados de las tareas en *completadas* pasan a *en revisión*, y desde esta columna existen dos opciones:
 - La tarea se ha completado con éxito por lo que pasaría a *cerradas*.
 - La tarea está incompleta o mal lo que haría que volviese a la columna *en proceso*.

- Dividir el proyecto en ciclos o iteraciones. Cada iteración representa una fase específica del proyecto, por ejemplo: investigación, diseño, implementación, pruebas, etc. La parte de implementación estará dividida a su vez por las distintas versiones de la aplicación (v0.1, v0.3, etc.), que incorporarán las nuevas funcionalidades previstas para esa versión.

- Planificar dichas iteraciones con las tareas a realizar, los objetivos que se deben lograr y los plazos establecidos. En cada una se limitará el trabajo en progreso del tablero Kanban, para así evitar una sobrecarga de trabajo. Al final de cada iteración, se realiza una entrega del trabajo realizado y se recibe una retroalimentación, a partir de la cual se realizarán los ajustes necesarios para la siguiente iteración.

- También es importante realizar una buena planificación de los riesgos asociados al proyecto y abordarlos de forma continuada a lo largo del mismo. Se deben de considerar en cada iteración e incorporar a la planificación todos aquellos que surjan durante el desarrollo.

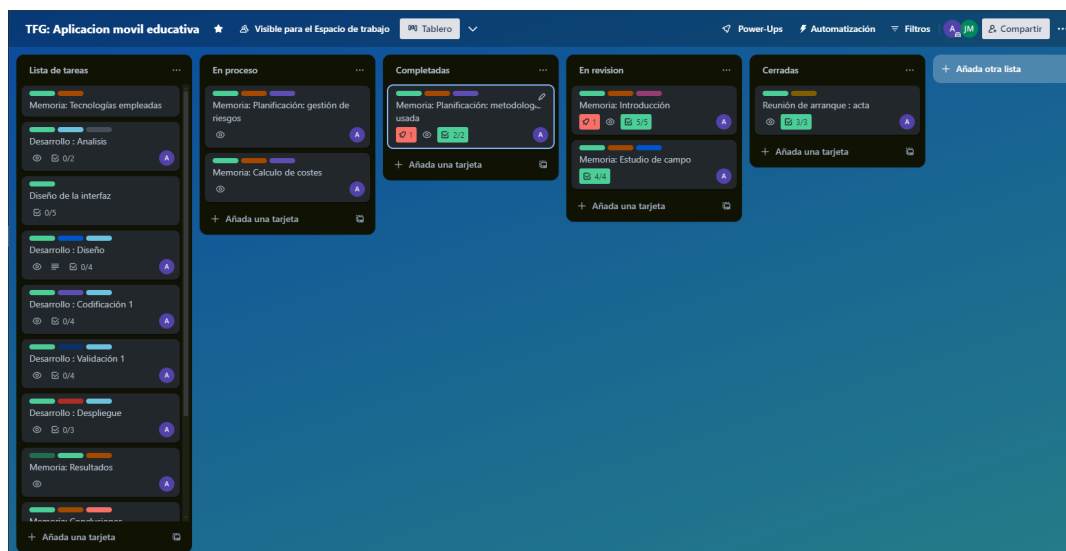


Figura 3.3: Tablero Kanban del proyecto

3.5. Planificación inicial

El proyecto fue propuesto a mediados del mes de octubre de 2023, partiendo de una idea inicial que habría que afinar y concretar más durante los posteriores meses. Durante los primeros meses de vida del proyecto el trabajo se centrará en la investigación y la planificación del proyecto.

Una vez se tuvo una idea sobre cómo orientar la aplicación dio comienzo la fase de análisis y diseño de la aplicación. Primero se definieron todos los requisitos, incluso si estos podían cambiar a lo largo del desarrollo. También se realizó un estudio para orientar la interfaz a usuarios de entre 7 y 11 años y sus padres. Tras esta investigación se creó una maqueta de la aplicación que sirvió como diseño inicial. Además de la interfaz también se realizaron los diagramas UML correspondientes al diseño del software de la aplicación, es decir, casos de uso, modelo de dominio y diagramas de secuencia.

A partir de febrero se comenzó con la parte de desarrollo de la aplicación. Previo a este desarrollo se realizó una formación por parte del alumno en tecnologías Android y en el lenguaje de programación Kotlin. Esta formación fue completamente online y estuvo a cargo del estudiante.

Respecto al seguimiento del desarrollo, este se realizó mediante iteraciones que terminaban con una reunión donde se mostraba el progreso y estado de la aplicación. Tras recibir una retroalimentación por parte del tutor se marcaban los objetivos de desarrollo para la siguiente iteración. Las pruebas y el mantenimiento de la aplicación se realizaron durante todo el desarrollo, aunque con especial importancia durante el tramo final.

CAPÍTULO 3. PLANIFICACIÓN DEL PROYECTO

El proyecto quedó cerrado y completado una vez se cumplieron las siguientes condiciones:

- Se ha cumplido con los objetivos del proyecto, tal y como se definieron en su comienzo o según se fueron incorporando.
- Se han realizado todos los entregables del proyecto, es este caso: la memoria, el código fuente y el APK de la versión final.
- El tutor ha revisado y aprobado los resultados del trabajo de fin de grado.
- Se han resuelto todos los problemas que han surgido durante el desarrollo, evitando que afecte a la calidad del producto final.
- El proyecto ha finalizado dentro de los plazos establecidos y respetando el presupuesto estimado.

Toda la planificación inicial se puede ver reflejada en el diagrama de Gantt del proyecto (3.4).

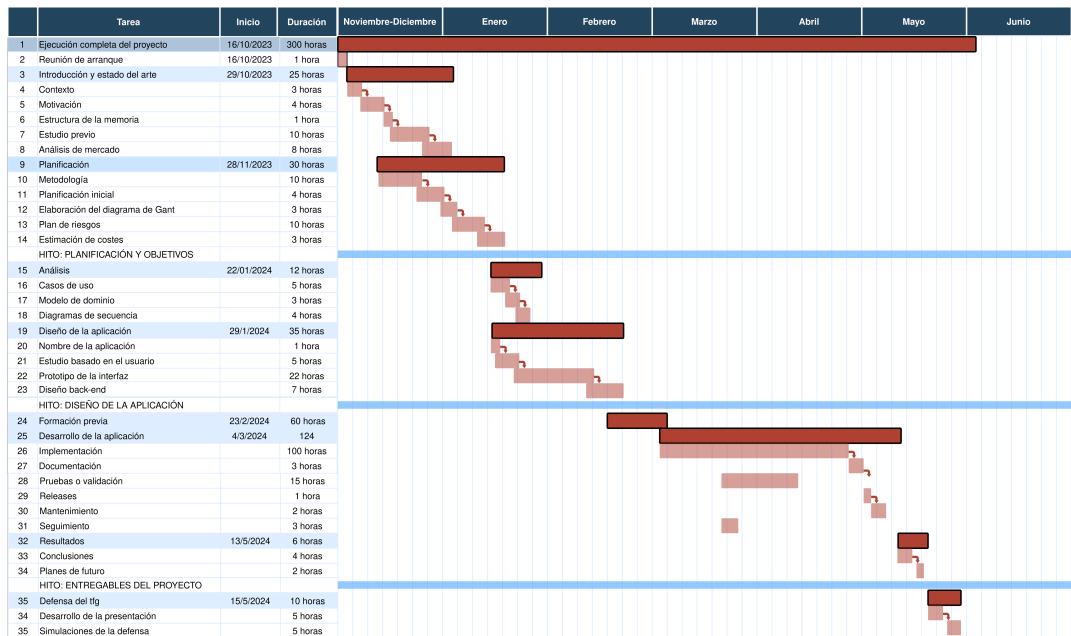


Figura 3.4: Diagrama de Gantt del proyecto

3.6. Plan de riesgos

Existen múltiples definiciones de un riesgo, por ejemplo, el PMBOK [34] lo describe como un evento inesperado o condición, que si ocurre, puede tener un efecto tanto positivo como

negativo en los objetivos del proyecto. Una buena planificación de riesgo busca reducir los efectos negativos y sacarle el máximo partido a los positivos. Es importante aclarar que solo se consideran los posibles problemas futuros, no aquellos que están presentes actualmente.

Una correcta planificación de riesgos debe estar formada por cuatro pasos o etapas:

1. Identificación de los riesgos.
2. Análisis de los riesgos y priorización de los mismos.
3. Planificación de los riesgos.
4. Monitorización de los riesgos.

Previo a la identificación de los riesgos se deben conocer las categorías de riesgos. La primera categoría son los riesgos de negocio, que afectan tanto a la viabilidad del proyecto como a su rentabilidad. En segundo lugar están los riesgos de proyecto, relacionados con la planificación del mismo y que afectan a su desarrollo y calidad final. Además según su naturaleza esto se pueden categorizar según múltiples aspectos, por ejemplo: de planificación, de alcance, de tecnología, de estructura...

Finalmente entre las diferentes estrategias que existen para identificar los riesgos se ha empleado la conocida como *brainstorming*. Se reúne al equipo, en este caso, tutor y alumno y se discuten los posibles riesgos y oportunidades que puedan surgir. A través de las reuniones, se identifican nuevos obstáculos y se desarrollan estrategias para mitigarlos. Este proceso se repite a lo largo del desarrollo, por lo que se pueden ir incorporando nuevos riesgos a la planificación.

3.6.1. Gestión inicial de riesgos

A continuación se detalla el plan de riesgos inicial del proyecto. Dado que trabajamos dentro de un marco ágil se pueden identificar nuevos riesgos a lo largo del desarrollo y en cada iteración se revisarán antes de comenzar con una nueva. Los nuevos riesgos quedarán reflejados en la subsección de *monitorización de los riesgos* 3.8.

En función de la probabilidad y el impacto de los riesgos se llevarán a cabo unas acciones u otras, detalladas en el análisis. En la matriz que se muestra a continuación (ver Tabla: 3.1) se pueden ver los diferentes planes de acción en función del peligro que representan.

A continuación se muestran los riesgos identificados durante la planificación, de la tabla 3.3 a la 3.10.

Probabilidad/ Impacto	Bajo	Medio	Alto
Bajo	Monitorizar	Monitorizar	Aplicar plan de mitigación
Medio	Monitorizar	Aplicar plan de mitigación	Mitigar y preparar un plan de contingencia
Alto	Aplicar plan de mitigación	Mitigar y preparar un plan de contingencia	Mitigar y preparar un plan de contingencia

Tabla 3.1: Matriz de riesgos.

ID	RSK001
Nombre	Baja por enfermedad u otros problemas de salud
Descripción	Caer enfermo durante la realización del TFG
Categoría	Equipo
Vulnerabilidad	Caer enfermo.
Amenaza	Perder productividad u horas de trabajo debido a la recuperación.
Probabilidad	Baja
Impacto	Alto
Riesgo Total	Medio
Acciones de mitigación	<ul style="list-style-type: none"> - Llevar una vida saludable. - Acudir a revisiones médicas ante indicios de enfermedad. - Planificar el proyecto teniendo en cuenta posibles bajas.
Acciones correctivas	<ul style="list-style-type: none"> - Acudir al médico. - Guardar reposo y no forzarse a trabajar si la situación no lo permite. - Retrasar las tareas hasta que el trabajador esté en buenas condiciones.

Tabla 3.2: Riesgo RSK001: Baja por enfermedad u otros problemas de salud.

ID	RSK002
Nombre	Practicas en empresa
Descripción	Se realizan las prácticas de empresa al mismo tiempo que se realiza el TFG.
Categoría	Equipo
Vulnerabilidad	Dedicar tiempo a las prácticas.
Amenaza	Se reduce el tiempo de dedicación al TFG.
Probabilidad	Media
Impacto	Medio
Riesgo Total	Medio
Acciones de mitigación	<ul style="list-style-type: none"> - Establecer un horario rígido para repartir el tiempo entre el trabajo y el TFG. - Realizar las prácticas previamente a la realización del TFG.
Acciones correctivas	<ul style="list-style-type: none"> - Pedir una prórroga en las prácticas. - Replanificar el tfg.

Tabla 3.3: Riesgo RSK002: Realización de las prácticas de empresa.

ID	RSK003
Nombre	Falta de formación en la tecnología
Descripción	Los desarrolladores tienen un escaso conocimiento de la tecnología y el lenguaje de programación.
Categoría	Tecnología
Vulnerabilidad	Falta de conocimientos
Amenaza	Los tiempos de desarrollo se ven incrementados
Probabilidad	Alta
Impacto	Medio
Riesgo Total	Alto
Acciones de mitigación	<ul style="list-style-type: none"> - Realizar una formación en la tecnología previamente al desarrollo. - Tener siempre a mano la documentación.
Acciones correctivas	<ul style="list-style-type: none"> - Pedir ayuda a algún experto. - Recortar el alcance del proyecto.

Tabla 3.4: Riesgo RSK003: Falta de formación en la tecnología

ID	RSK004
Nombre	Planificación poco realista del proyecto
Descripción	El tiempo estimado para completar el proyecto es demasiado optimista y poco realista.
Categoría	Planificación
Vulnerabilidad	Mala planificación
Amenaza	No se llega a completar el proyecto en los tiempos estimados
Probabilidad	Media
Impacto	Alto
Riesgo Total	Alto
Acciones de mitigación	- Emplear un marco de trabajo ágil mediante el que se realizará una revisión de la planificación en cada iteración.
Acciones correctivas	- Dedicar más horas de trabajo. - Recortar el alcance del proyecto. - Retrasar la fecha de finalización.

Tabla 3.5: Riesgo RSK004: Planificación poco realista del proyecto

ID	RSK005
Nombre	Mal diseño de la interfaz
Descripción	Se ha diseñado una interfaz de usuario inadecuada y poco intuitiva.
Categoría	Tareas
Vulnerabilidad	Mala interfaz
Amenaza	La aplicación fracasa en el mercado debido a que la interfaz tiene un mal diseño.
Probabilidad	Media
Impacto	Medio
Riesgo Total	Medio
Acciones de mitigación	- Emplear la metodología iterativa basada en prototipos. - Realizar un estudio previo en el diseño basado en los usuarios. - Aplicar buenas prácticas.
Acciones correctivas	- Rehacer la interfaz.

Tabla 3.6: Riesgo RSK005: Mal diseño de la interfaz

ID	RSK006
Nombre	Cambios de requisitos
Descripción	Los requisitos del proyecto pueden cambiar, añadirse nuevos o eliminarse otros.
Categoría	Planificación
Vulnerabilidad	Cambios de los requisitos
Amenaza	Reajuste del alcance y las actividades planificadas.
Probabilidad	Media
Impacto	Alto
Riesgo Total	Alto
Acciones de mitigación	<ul style="list-style-type: none"> - Emplear una metodología ágil. - Comunicación transparente entre el alumno y el profesor. - Evaluar adecuadamente el nivel de impacto del cambio.
Acciones correctivas	<ul style="list-style-type: none"> - Reajustar el alcance del proyecto. - Revisar la planificación.

Tabla 3.7: Riesgo RSK006: Cambios de requisitos

ID	RSK007
Nombre	Gold plating
Descripción	Se han desarrollado funciones que van más allá de los requisitos iniciales y que no son necesarias para los objetivos del proyecto.
Categoría	Tareas
Vulnerabilidad	Desarrollar funcionalidades innecesarias
Amenaza	Pueden quitar tiempo de desarrollar funcionalidades más importantes.
Probabilidad	Baja
Impacto	Medio
Riesgo Total	Bajo
Acciones de mitigación	<ul style="list-style-type: none"> - Refinar las tareas del tablero. - Valorar si la función es realmente necesaria.
Acciones correctivas	<ul style="list-style-type: none"> - Bloquear la tarea. - Abandonar la tarea.

Tabla 3.8: Riesgo RSK007: Gold plating

ID	RSK008
Nombre	Falta de acceso a recursos
Descripción	Imposibilidad de acceder a ciertos recursos o software por falta de licencia.
Categoría	Recursos
Vulnerabilidad	Falta de licencias
Amenaza	Puede retrasar o impedir alcanzar los objetivos del proyecto.
Probabilidad	Baja
Impacto	Alto
Riesgo Total	Medio
Acciones de mitigación	- Solicitar licencias a la universidad - Invertir parte del presupuesto en las licencias
Acciones correctivas	- Buscar alternativas open source

Tabla 3.9: Riesgo RSK008: Falta de acceso a recursos

ID	RSK009
Nombre	Mala coordinación con el tutor
Descripción	Una mala coordinación puede afectar al desarrollo del proyecto
Categoría	Equipo
Vulnerabilidad	Mala comunicación entre alumno y profesor
Amenaza	Mala orientación del proyecto y fracaso del mismo
Probabilidad	Baja
Impacto	Alto
Riesgo Total	Medio
Acciones de mitigación	- Establecer una buena comunicación entre el alumno y el tutor. - Establecer reuniones periódicas para aclarar las dudas y problemas que surjan.
Acciones correctivas	- Realizar reuniones extra o incrementar la frecuencia con la que se realizan.

Tabla 3.10: Riesgo RSK009: Mala coordinación con el tutor

3.6.2. Gestión inicial de oportunidades

Al igual que pueden manifestarse riesgos a lo largo del desarrollo también pueden surgir oportunidades. Para sacarle el máximo partido es importante establecer también una planificación de las oportunidades. A continuación se muestran las oportunidades identificadas en la planificación, de la tabla 3.11 a la 3.14.

ID	OP001
Nombre	Emplear un marco de trabajo ágil junto a un desarrollo iterativo.
Descripción	Realizar iteraciones que finalicen con un entregable para validar.
Categoría	Planificación
Oportunidad	Se reducen los riesgos de que el producto final no sea lo que se deseaba y se puede ir encaminando hacia aquello que sí.
Probabilidad	Alta
Impacto positivo	Medio
Nivel de oportunidad	Alto
Acciones de aprovechamiento	<ul style="list-style-type: none"> - Ajustar los requisitos en cada iteración. - Entregas más frecuentes. - Dividir el proyecto en partes más pequeñas.

Tabla 3.11: Oportunidad OP001: Emplear un marco de trabajo ágil

ID	OP002
Nombre	Intercambio de ideas.
Descripción	Aunque el TFG es un trabajo individual no impide intercambiar opiniones e ideas con el tutor o compañeros.
Categoría	Personal.
Oportunidad	Nuevas ideas y puntos de vista.
Probabilidad	Baja
Impacto positivo	Medio
Nivel de oportunidad	Bajo
Acciones de aprovechamiento	<ul style="list-style-type: none"> - Tomar notas de las opiniones. - No tomar la opinión propia como la mejor, valorar otras opciones.

Tabla 3.12: Oportunidad OP002: Intercambio de ideas

ID	OP003
Nombre	Hacer pruebas con usuarios reales.
Descripción	Oportunidad de hacer pruebas con usuarios reales de la aplicación.
Categoría	Proyecto
Oportunidad	Validar la aplicación con datos de usuarios.
Probabilidad	Media
Impacto positivo	Medio
Nivel de oportunidad	Medio
Acciones de aprovechamiento	- Establecer un plan de pruebas. - Recopilar retroalimentación mediante la observación o mediante cuestionarios.

Tabla 3.13: Oportunidad OP003: Hacer pruebas con usuarios reales

ID	OP004
Nombre	Colaboración de un experto.
Descripción	Contar con la colaboración de un experto en el campo de la educación.
Categoría	Proyecto
Oportunidad	Aporte de ideas de un experto.
Probabilidad	Baja
Impacto positivo	Alto
Nivel de oportunidad	Medio
Acciones de aprovechamiento	- Tener preparado un prototipo de la aplicación y preguntas sobre posibles funcionalidades que añadir.

Tabla 3.14: Oportunidad OP004: Colaboración de un experto

3.7. Estimación de costes

En la estimación de los costes del proyecto se debe tener en cuenta el coste de la mano de obra, del material de trabajo, software necesario para la elaboración del proyecto así como el coste de formación previa del equipo. Debe tenerse en cuenta que el proyecto se desarrolla bajo un contexto académico, por lo que los costes serán en todo momento estimados y no reales.

- Respecto al coste de la mano de obra, el equipo de trabajo de este proyecto esta formado por un único desarrollador de software, en concreto un desarrollador de aplicaciones Android. En España, a día 14 de febrero del 2024, el sueldo medio de un desarrollador Android es de 2693,2€ netos al mes según el portal de búsqueda de empleo indeed.com

[35]. Es muy importante tener también en cuenta las contribuciones que se deben realizar a la seguridad social, se puede ver el calculo en la tabla 3.15 a continuación.

Sueldo neto mensual (12 pagas)	2693,2€
Sueldo neto anual	32 318,9€
Tipo de retención sobre la nómina	20,94 %
Cuotas a la Seg.Social	2822,5€
Retenciones por IRPF	9307,6€
Sueldo bruto anual	44 449,0€

Tabla 3.15: Calculo de retenciones y salario

Dado que el desarrollo del proyecto está estimado en 300 horas, que equivaldría aproximadamente a dos meses trabajando a tiempo completo, el coste de mano de obra quedaría estimado en 6280€.

- El hardware y material de trabajo consiste en un ordenador de sobremesa que en su día costó 950€ con una vida útil estimada de entre 5 a 7 años(84 meses). Aunque la estimación del proyecto es de 300 horas la duración del mismo es mayor, debido a como se planificó la distribución de estas horas, resultando en que el equipo de trabajo se utilizará durante un mínimo seis meses y quedando la estimación del coste hardware en 67,86€.
- En lo respectivo al software, para la realización de este proyecto se empleará siempre un software gratuito o del que se disponga de una licencia académica. Sin embargo, la plataforma Overleaf a partir de enero de 2024 comenzó a limitar los tiempo de compilación a menos que se contratase la versión premium. El coste de esta licencia para los meses necesarios para completar la memoria fue de 18€.
- La formación previa consistirá en una formación online impartida a través de la plataforma Udemy cuyo coste es de 14,99€.
- Finalmente también se deben considerar los costes fijos de la oficina o lugar de trabajo, donde se incluyen los costes de luz, gas y acceso a internet. Estos varían en función del mes y el consumo, por lo que se emplea una media de 130€ para el calculo. Como el proyecto durará unos 6 meses, los costes fijos serán de 780€.

Sumando todos los costes estimados previamente nos quedaría un coste estimado del proyecto de 7802,85€ (ver tabla 3.16).

Mano de obra	6280€
Coste del hardware	67,86€
Coste del software	18€
Formación	14,99€
Costes fijos	780€
Total	7160,85€

Tabla 3.16: Calculo del coste del proyecto

3.8. Monitorización de los riesgos

Durante el proyecto aparecieron algunos de los riesgos contemplados en el plan de riesgos original junto a algunos nuevos. De entre los riesgos estimados inicialmente algunos de los más repetidos fueron:

- **Falta de formación en la tecnología:** la falta de experiencia en proyectos parecidos a incrementado en ocasiones los tiempo necesarios para completar algunas tareas. Algunos de los peores fueron el desarrollo de la funcionalidad de la cámara o habituarse a trabajar con *RecycleViews*.
- **Falta de acceso a recursos:** este riesgo se ha manifestado en algunas ocasiones en menor medida, como por ejemplo al querer usar una funcionalidad de *Firebase* y que esta fuese de pago, pero sobretodo la más destacada fue la del editor *LaTeX* de *Overleaf*, el cual pasó a ser de pago.
- **Planificación poco realista del proyecto:** este riesgo es de los más repetidos a lo largo de proyecto, el impacto de otros riesgo deriva en este. Siempre que se han producido cambios respecto a la planificación original se ha intentado incrementar las horas dedicadas por encima de recortar el alcance del proyecto.

Respecto a los riesgos no estimados en la planificación inicial y que han ido surgiendo durante el desarrollo tenemos los siguientes:

ID	RSK010
Nombre	Preparación de exámenes
Descripción	El tiempo dedicado a preparar exámenes redujo el tiempo disponible para el proyecto.
Categoría	Planificación
Vulnerabilidad	Carga de trabajo adicional.
Amenaza	Retrasa la finalización de tareas y afecta al progreso general.
Impacto	Medio
Acciones correctivas	<ul style="list-style-type: none"> - Recuperar el trabajo perdido durante las posteriores semanas. - Emplear tiempo del “colchón” o margen de tiempo estimado.

Tabla 3.17: RSK010: Preparación de exámenes

ID	RSK011
Nombre	Estimación incorrecta de las tareas
Descripción	Planificar en base a estimaciones incorrectas del tiempo necesario para implementar ciertas funcionalidades.
Categoría	Tareas
Vulnerabilidad	No se completan ciertas tareas en una iteración y estas se tienen que mover a la siguiente.
Amenaza	Afectar a la planificación inicial del proyecto.
Impacto	Medio
Acciones correctivas	- Reajustar las horas dedicadas - Ajustar la planificación según sea necesario

Tabla 3.18: RSK011: Estimación incorrecta de las tareas

ID	RSK012
Nombre	<i>Bugs</i> o errores en el código
Descripción	Aparición de errores en el código que necesitan ser resueltos.
Categoría	Calidad del software
Vulnerabilidad	Falta de experiencia y de pruebas.
Amenaza	Puede causar fallos, pérdida de datos y producir una mala experiencia para el usuario.
Impacto	Alto
Acciones de mitigación	- Revisiones por pares durante las reuniones. - Pruebas unitarias y de integración.
Acciones correctivas	- Identificar y corregir los errores. - Realizar revisiones y pruebas adicionales después de cada corrección

Tabla 3.19: RSK012: *Bugs* o errores en el código

3.9. Comparación respecto a la planificación inicial

Si que se ha producido una cierta desviación respecto a la planificación inicial, sobre todo durante la fase de desarrollo, principalmente a causa de los riesgos mencionados en secciones anteriores. Esta desviación no ha tenido consecuencias graves como reducir el alcance o retrasar la entrega, debido por una parte a que la mayoría de los riesgos se podían corregir incrementando el número de horas dedicadas y, por otra parte, gracias a que el proyecto contaba con cierto margen de error (ver figura 3.5).

Otra desviación importante respecto a la planificación inicial fue no encontrar un grupo de usuarios adecuado para probar la maqueta de la interfaz previamente al desarrollo. Debido a que el desarrollo necesitaba comenzar cuanto antes se confió en que siguiendo buenas pautas

CAPÍTULO 3. PLANIFICACIÓN DEL PROYECTO

de diseño sería suficiente. Más adelante sí que se pudo hacer pruebas de validación con la aplicación en un estado avanzado de desarrollo. Los resultados de dichas pruebas fueron positivos debido a que los únicos fallos detectados se podían corregir fácilmente sin hacer grandes cambios en el código y el diseño (ver tabla 6.2).

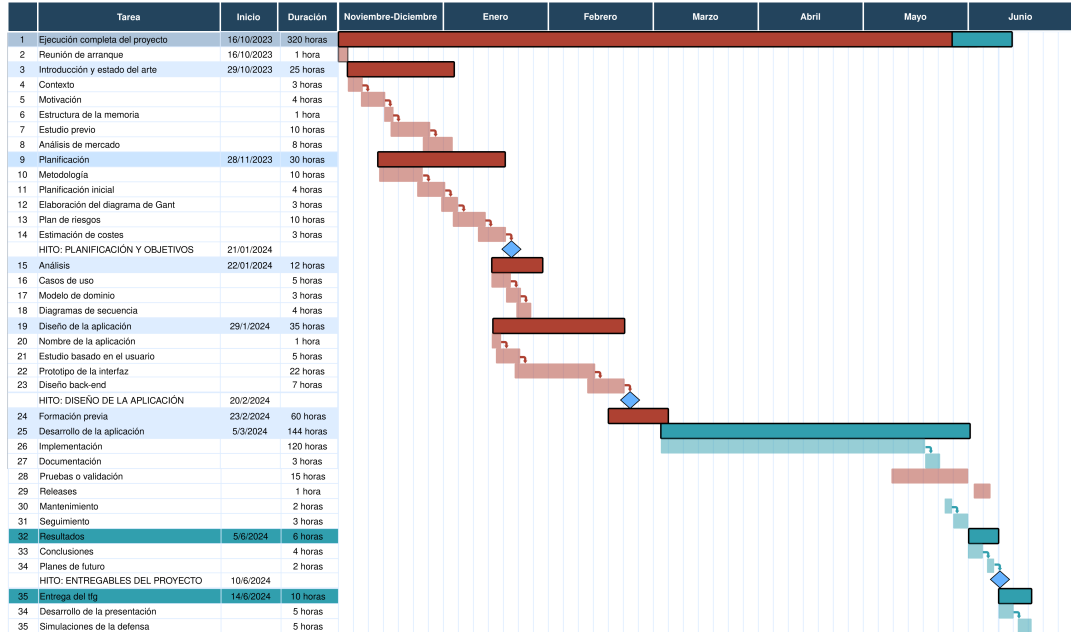


Figura 3.5: Desviación respecto a la planificación. Los cambios están resaltados en otro color.

Capítulo 4

Tecnologías

4.1. Herramientas de comunicación

Microsoft Teams

Microsoft Teams es una aplicación de colaboración creada para el trabajo híbrido y la coordinación de equipos. Tiene funciones de reunión en línea, organización de eventos, videoconferencia y documentos compartidos. Es una de las herramientas más usadas en el mundo laboral y, además, está financiada por la Universidad de Valladolid.

Outlook

Aplicación de correo electrónico ofrecida por Microsoft y empleada para los correos dentro de la institución educativa. Esta herramienta se empleará para correos más formales y la comunicación en las primeras fases del proyecto cuando todavía no se han establecido unas reuniones periódicas.

4.2. Herramientas para el análisis, diseño y documentación

Overleaf

Overleaf es un editor colaborativo de LaTeX basado en la nube que se utiliza para escribir, editar y publicar documentos científicos. Se asocia con una amplia gama de editoriales científicas y ofrece una serie de plantillas para adaptarse al estilo de documento que se desea realizar [36].

Figma

Figma es un editor de gráficos vectorial y una herramienta de generación de prototipos basada en la web. Se dibuja cada pantalla de una aplicación o sitio web y se establece una conexión entre ellas. Es decir, se modela el comportamiento del usuario. El resultado es un prototipo del producto, que es fácil de presentar, editar y validar con el cliente [37][38].

Draw.io

Draw.io es una plataforma en línea que permite crear diagramas y gráficos de manera colaborativa y fácil. Permite la creación de una variedad de diagramas UML, como diagramas de flujo, modelos de dominio, diagramas de casos de uso y muchos otros tipos de representaciones visuales. Puede integrarse con diversas plataformas y servicios y además ofrece plantillas predefinidas para diferentes tipos de diagramas[39][40].

Canva

Canva es una web de diseño gráfico y composición de imágenes que ofrece herramientas online para crear diseños. Se ha usado para crear algunos elementos visuales como el logo de la aplicación.

4.3. Herramientas para la gestión y control del proyecto

Trello

Trello es una aplicación online diseñada para la gestión de actividades y la administración de proyectos. Este programa está optimizado para organizar información en formatos visuales simples que facilitan la realización de tareas y el cumplimiento de objetivos[41][42].

Git

Git es un sistema de control de versiones de software libre que se utiliza para el seguimiento de cambios en el código durante el desarrollo de software. Desde su creación en 2005 por *Linus Torvalds* se ha convertido en uno de los sistemas de control de versiones más populares y utilizados de la industria[43].

GitHub

Github es un repositorio online y gratuito que permite gestionar proyectos y controlar versiones del código. Es muy utilizado por desarrolladores de todo el mundo tanto para proteger el código en la nube como para gestionarlo. Además permite compartirlo con otros desarrolladores de forma sencilla[44][45].

SourceTree

SourceTree[46] es una herramienta de gestión de versiones que facilita la interacción con repositorios *Git*. Es desarrollado por Atlassian y proporciona una interfaz gráfica intuitiva para realizar operaciones de control de versiones. Algunas de las características que incluye son: visualización de ramas, realización de fusiones (*merges*) y resolución de conflictos, guardar código de forma temporal (*stash*), gestión de repositorios remotos y realizar operaciones *Git* de forma simplificada.

4.4. Herramientas para el desarrollo del proyecto

Android Studio

Android Studio es un entorno de desarrollo integrado diseñado por Google para el desarrollo de aplicaciones Android. Este software proporciona herramientas y recursos para facilitar la creación, depuración y optimización de aplicaciones móviles que se ejecutan en un sistema operativo Android[47]. Entre las ventajas que ofrece se pueden destacar[48]:

- **Emulador de Android integrado:** incluye un emulador de Android que permite probar las aplicaciones en diferentes versiones y dispositivos.
- **Sistema de construcción flexible:** emplea Gradle como sistema de dependencias, lo que facilita su gestión y personalización.
- **Soporte para Kotlin:** Android Studio soporta el desarrollo en Kotlin, un lenguaje que se posiciona por encima de Java en lo respectivo a desarrollar aplicaciones Android en los últimos años.
- **Plantillas:** incluye plantillas que ayudan a crear componentes comunes como actividades, fragmentos y servicios.

Firebase

Firebase es una plataforma en la nube proporcionada por Google para el desarrollo de aplicaciones web y móviles. Esta plataforma ofrece una amplia gama de herramientas y servicios que facilitan el desarrollo, la gestión y la mejora de las aplicaciones. Entre las funcionalidades que se emplean de Firebase está:

- **Base de datos:** almacena y sincroniza datos entre los usuarios en tiempo real.
- **Autenticación:** proporciona métodos para autenticar a los usuarios mediante correo, teléfono y servicios de terceros.
- **Análisis:** incluye herramientas para analizar el comportamiento de los usuarios y el rendimiento de la aplicación.

- **Almacenamiento backend:** permite almacenar archivos de los usuarios como imágenes o música en la nube.
- **App Check:** protege las API y recursos del backend, verificando las solicitudes para evitar ataques maliciosos.

Firebase se integra muy fácilmente con Android gracias a que ambos los desarrolla Google, lo que lo convierte en una opción versátil para construir y escalar aplicaciones de manera eficiente [49].

Capítulo 5

Análisis

5.1. Requisitos

Los requisitos son descripciones detalladas y claras de las características y funcionalidades que se esperan de la aplicación. Proporcionan una base para la planificación, desarrollo y evaluación de la misma.

5.1.1. Requisitos funcionales

Los requisitos funcionales describen las funcionalidades principales que debe tener la aplicación. A continuación se listan los requisitos funcionales identificados tras estudiar como orientar la educación, las estrategias recogidas en la sección **2.1.2**, la investigación de aplicaciones parecidas y las reuniones con el tutor:

- **RF-01** La aplicación deberá permitir a los adultos registrarse con nombre de la familia, nombre de usuario, e-mail y contraseña.
- **RF-02** La aplicación deberá permitir acceder a la familia mediante e-mail y contraseña.
- **RF-03** La aplicación deberá permitir a los adultos registrar a los menores de la familia mediante nombre de usuario y contraseña opcional.
- **RF-04** La aplicación deberá permitir a los adultos asignar la propina de los menores.
- **RF-05** La aplicación deberá ofrecer diferentes formatos de propina.
- **RF-06** La aplicación deberá permitir a los adultos eliminar los datos de un menor del sistema o darse de baja.
- **RF-07** La aplicación deberá ser capaz de llevar un registro de los movimientos de dinero.

- **RF-08** La aplicación deberá permitir a los adultos monitorizar los movimientos de dinero que hagan los menores.
- **RF-09** La aplicación deberá permitir a los adultos crear diferentes tareas para la familia.
- **RF-10** La aplicación deberá permitir a los adultos poner fecha límite y prioridad a las tareas.
- **RF-11** La aplicación deberá permitir a los adultos añadir una recompensa a las tareas que desee.
- **RF-12** La aplicación deberá permitir acceder a los menores en la aplicación simplemente mediante su perfil o perfil y contraseña.
- **RF-13** La aplicación deberá mostrar a los menores el dinero que tienen disponible y sus movimientos de dinero.
- **RF-14** La aplicación deberá permitir a los menores registrar gastos.
- **RF-15** La aplicación deberá permitir crear nuevos perfiles de adultos o tutores legales dentro de la familia.
- **RF-16** La aplicación deberá permitir marcar tareas como completadas.
- **RF-17** La aplicación no deberá eliminar una tarea completada hasta que los adultos la revisen y marquen como realmente completada.
- **RF-18** La aplicación deberá permitir a los menores crear objetivos de ahorro.
- **RF-19** La aplicación deberá permitir a los menores añadir la cantidad que deseen de dinero a un objetivo de ahorro.
- **RF-20** La aplicación deberá permitir a los menores jugar a pequeños juegos educativos.
- **RF-21** La aplicación deberá permitir a los usuarios restablecer la contraseña.
- **RF-22** La aplicación deberá verificar el correo electrónico de los usuarios.
- **RF-23** La aplicación deberá permitir cerrar sesión a los usuarios.
- **RF-24** La aplicación deberá mostrar las condiciones de uso y privacidad.
- **RF-25** La aplicación deberá permitir modificar el nombre de un miembro, nombre de familia, correo y contraseña.

5.1.2. Requisitos no funcionales

Los requisitos no funcionales describen las funcionalidades que no son necesarias para la consecución de los objetivos del proyecto, pero que mejoran la calidad del producto final. A continuación se listan los requisitos no funcionales identificados:

- **RNF-01** La aplicación permitirá la opción de enviar notificaciones por e-mail sobre la actividad de los menores.
- **RNF-02** La aplicación permitirá la opción de notificaciones *push* sobre la actividad de los menores.
- **RNF-03** La aplicación permitirá tener iconos personalizables para todos los perfiles.
- **RNF-04** La aplicación tendrá *widgets*.
- **RNF-05** La aplicación tendrá un historial con las tareas completadas.
- **RNF-06** La aplicación permitirá a los adultos calcular la economía familiar.
- **RNF-07** La aplicación tendrá la opción de *autologin*.

5.2. Casos de uso

Los casos de uso representan las interacciones que tienen los actores (usuarios o sistema externo) con nuestra aplicación para lograr un objetivo en concreto. Como hemos visto anteriormente nuestra aplicación tendrá dos tipos de usuarios: *adultos* y *menores*. Cada perfil tendrá sus propios casos de uso, aunque existen algunas interacciones comunes a ambos perfiles. Con el objetivo de simplificar los diagramas y mejorar la claridad se ha creado un tercer actor *usuario* donde se recogerán aquellas operaciones comunes a todos los tipos de perfil.

El resultado son tres diagramas de casos de uso (figuras 5.1, 5.2 y 5.3). En el primer diagrama se pueden ver la operaciones comunes de ambos perfiles, el segundo muestra los casos de uso relativos a los adultos y el tercero los relativos a los menores.

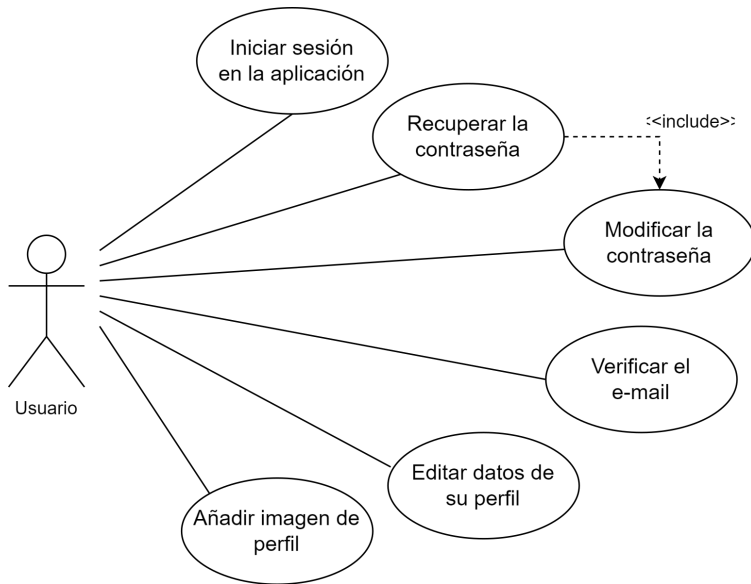


Figura 5.1: Casos de uso para todo tipo de usuario.



Figura 5.2: Casos de uso para los adultos.

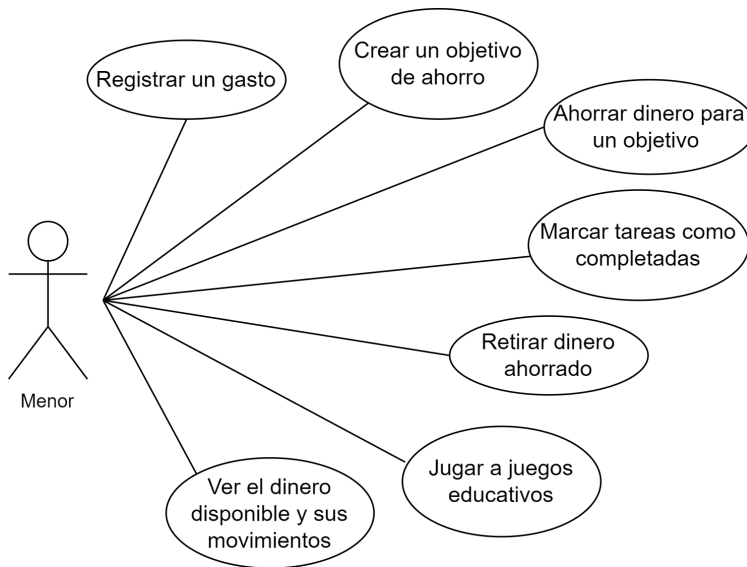


Figura 5.3: Casos de uso para los menores.

5.3. Modelo de dominio

En este apartado se muestra el modelo de dominio de la aplicación (ver figura 5.4) y se detallan las relaciones entre clases.

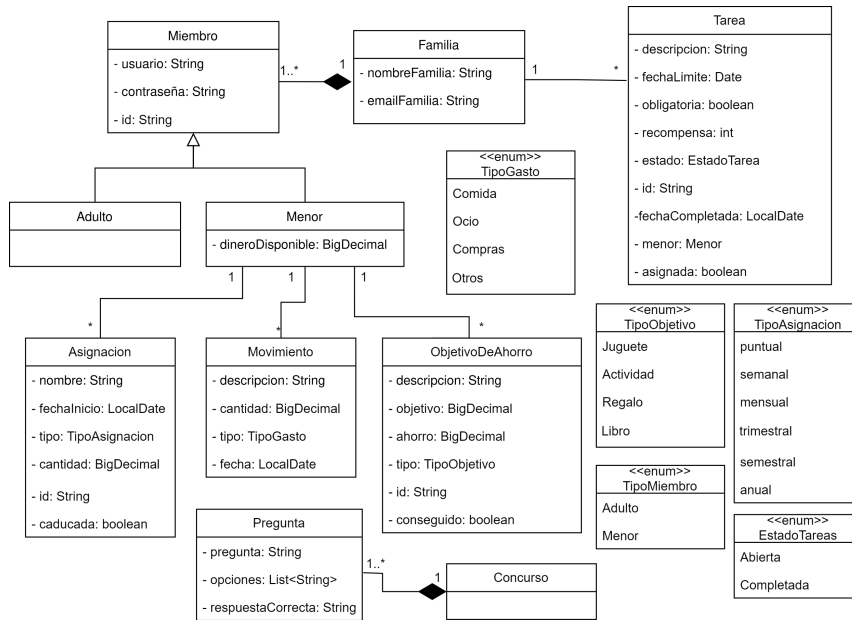


Figura 5.4: Modelo de dominio

En este modelo se conceptualizan:

- **Familia:** una familia esta compuesta por sus diferentes miembros y sin estos no tiene razón de existir. Tiene un nombre identificativo y un correo electrónico con el que se registran e identifican las familias en la aplicación.
- **Miembros:** los miembros de una familia pueden ser de dos tipos, perfil adulto o de un menor. Ambos tienen un nombre de usuario y contraseña, aunque esta última es opcional debido a que solo bloquea el acceso al perfil (se recordará en todo momento que es recomendable). A mayores los menores tienen un atributo con el dinero que tienen disponible. Las operaciones que puede hacer cada perfil también son diferentes.
- **Asignación:** la asignación o propina es creada por un perfil adulto para un menor de la familia en concreto. La fecha de inicio marca el momento en el que se inicia la asignación y esta puede ser de varios tipos, cambiando la frecuencia con la que se realiza.
- **Movimientos:** movimientos de dinero que han tenido los menores de la familia. Estos deben de apuntar los gastos como medio para acostumbrarles a llevar un registro y control de ellos. Las entradas de dinero se actualizarán automáticamente.
- **Objetivos:** meta de ahorro para los menores. Como forma de fomentar el ahorro y de planificar futuras compras, se crean estos objetivos. Mostrarán cual es el objetivo final y cuanto llevan ahorrado (emulando una hucha). En cualquier momento podrán retirar todo el dinero que tenían guardado y borrar dicho objetivo, aunque se intentará enseñar que lo ideal es que lleguen antes a esos objetivos.

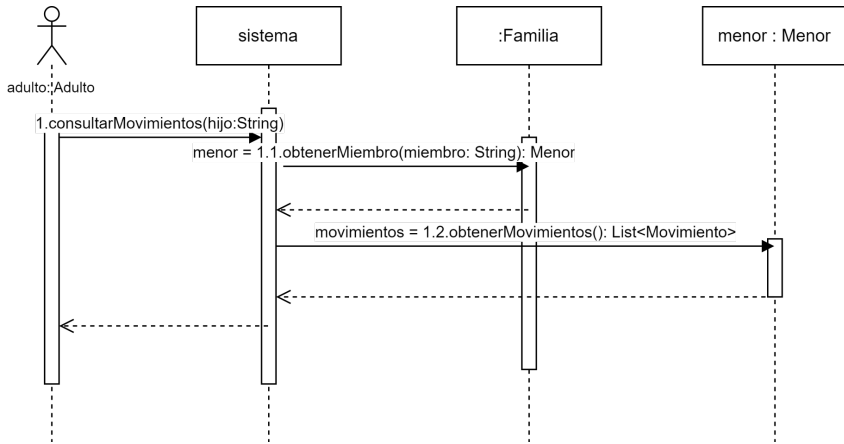


Figura 5.6: Diagrama de secuencia del CU “Ver movimientos de dinero de los menores”

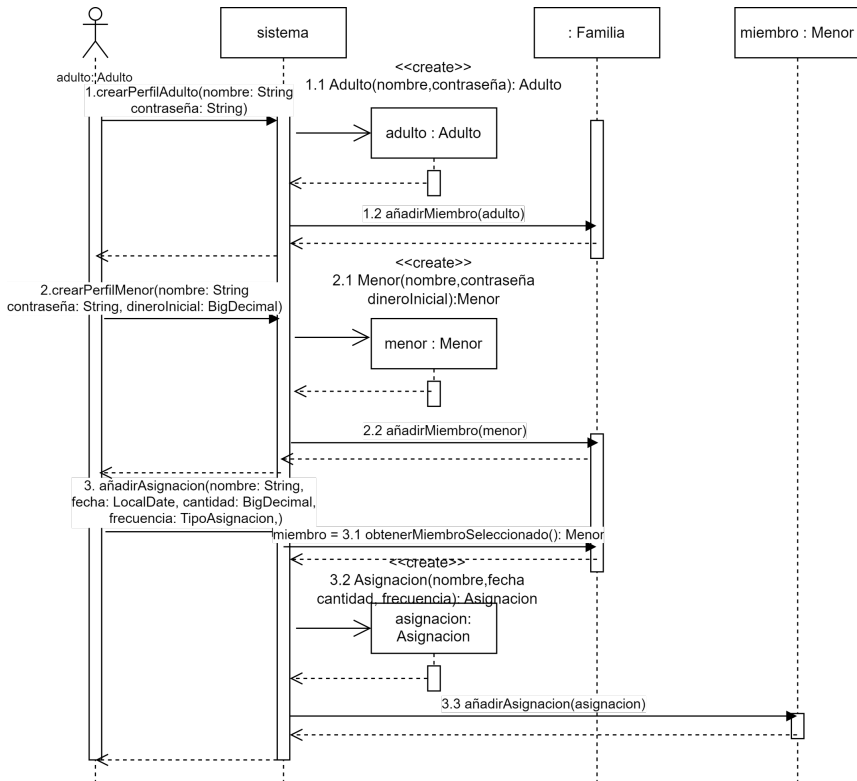


Figura 5.7: Diagrama de secuencia del CU “Registrar a un miembro de la familia”

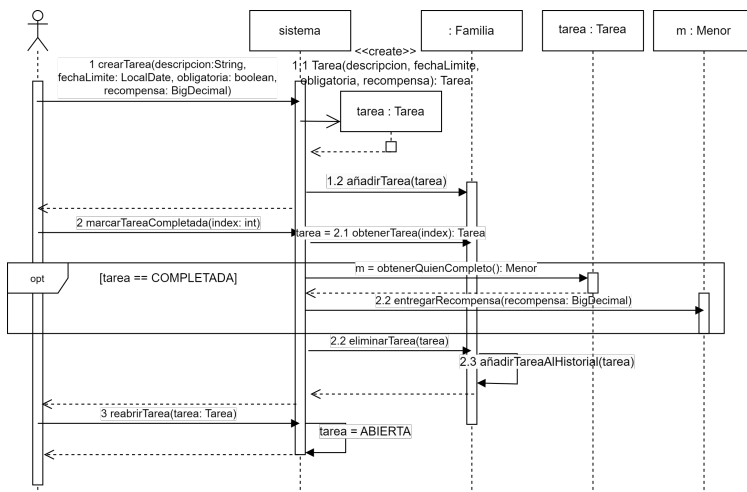


Figura 5.8: Diagrama de secuencia del CU “Crear una tarea”

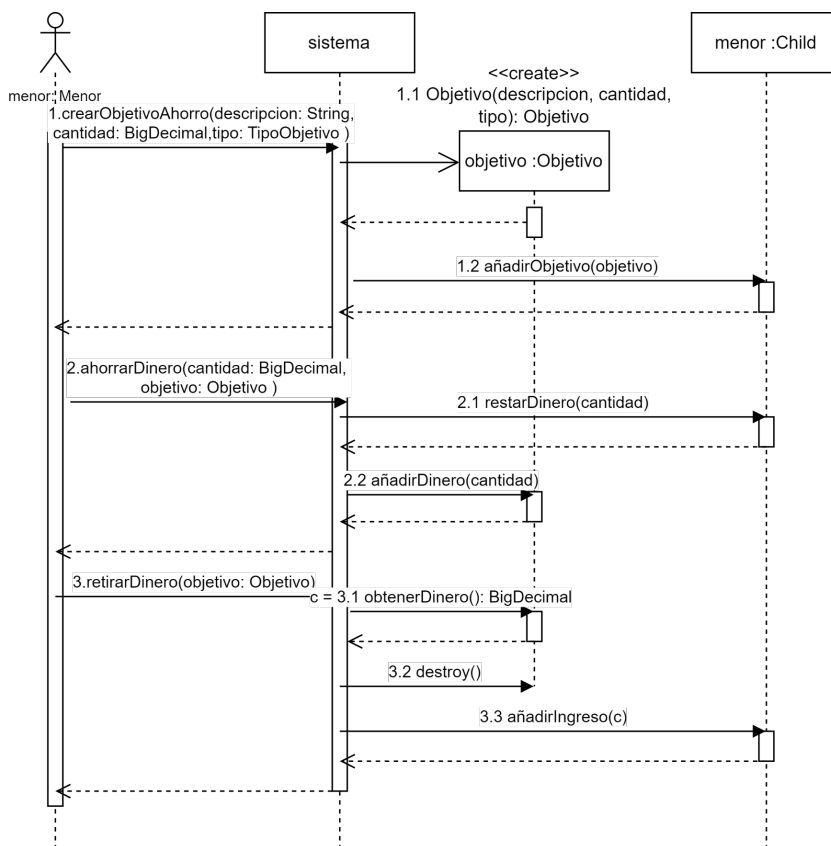


Figura 5.9: Diagrama de secuencia del CU “Crear un objetivo de ahorro”

Capítulo 6

Diseño

6.1. Diseño de la interfaz de usuario

En este capítulo, se detalla el proceso de diseño de interfaz llevado a cabo a lo largo del proyecto. Un buen diseño de la interfaz de usuario mejora la experiencia del usuario y la eficiencia del sistema, además de algunos beneficios entre los que destacar:

- Crea una experiencia más positiva para el usuario.
- Aumenta el tiempo que pasan los usuarios dentro de la aplicación.
- Incrementa la productividad y la seguridad de la aplicación, mejorando la comprensión y uso adecuado de la misma.
- Reduce costes de desarrollo al evitar demasiados cambios o correcciones.

En resumen, el éxito de una aplicación recae en gran medida en si su interfaz de usuario tiene un diseño bueno y apropiado [50].

6.1.1. Investigación de un diseño apropiado

Para que la aplicación tenga una interfaz bien diseñada primero se debe pensar en el público al que va dirigida. Un estudio de Nielsen Norman Group [51] llegó a la conclusión de que no existe un único marco para diseñar aplicaciones para niños, estas se deben adaptar en función de tres grupos: niños pequeños (3–5 años), de rango intermedio (6–8 años) y mayores (9–12 años). Cada grupo presenta diferentes comportamientos, capacidades físicas y cognitivas, y además los niños se vuelven más hábiles con la tecnología a medida que crecen.

Nuestro público se encuentra en un rango entre intermedio y mayores (7-11 años), por lo que debemos de emplear las características que mejor se adapten entre ambos grupos para

nuestra aplicación. Es importante destacar que los usuarios jóvenes reaccionan negativamente al contenido diseñado para niños que están grados educativos por debajo de ellos, es decir, si la interfaz es demasiado infantil, por ejemplo con demasiado dibujos o animaciones, pueden sentir rechazo hacia ella y dejar de usarla.

Dado que nuestra aplicación tendrá dos tipos de interfaz, una para adultos y otra para niños, se recogen una serie de comportamientos de cada grupo a ciertas características, todo ello para posteriormente adaptar la interfaz [52]. Algunas características son comunes a ambos grupos mientras que en otras se difiere un poco (ver Tabla 6.1).

Característica	Niños	Adultos
Primeras impresiones de la aplicación	Juzga rápido y la abandona si no es buena	Juzga rápido y la abandona si no es buena
Navegación redundante	Muy confusa	Algo confusa
Botón de retroceso	Muy usado	Muy usado
Lectura	Evitan si es posible	Mediante escaneo
Tamaño de la fuente	Mínimo 12	Mínimo 12 (más para gente mayor)
Scrolling	Evitarlo	Algo
Uso de gestos	Acciones simples	Bien conocidos
Objetivos principales del uso de aplicaciones	Entretenimiento	Variados
Comportamiento al explorar	Intentan todas las opciones posibles	Se mantienen en el camino principal
Anuncios	Pueden confundirlo con contenido de la aplicación	Lo identifican pero les molesta
Distracciones externas	Muy propenso	Algo propenso
Contraste de colores	Ayuda a localizar elementos	Puede saturar

Tabla 6.1: Descripción de comportamientos/características en niños y adultos.

6.1.2. Guías de diseño seguidas

Nuestro diseño debe seguir buenas practicas del diseño de interfaces [53], que en algunos casos se deben adaptar en función del grupo al que va dirigido. Las principales características para una buena interfaz son:

- Retroalimentación continua, hacer siempre visible el estado del sistema. El usuario debe saber en todo momento lo que está sucediendo.
- Se debe emplear un lenguaje común para todo el mundo, evitando tecnicismos.
- El usuario debe tener control pleno y libertad para hacer lo que desee.
- El diseño debe ser consistente y definir una serie de estándares comunes a toda ella, como la paleta de colores y la tipografía.

- Prevenir los errores ya desde el diseño de la interfaz y hacerlos visibles cuando estos ocurran.
- El usuario no tiene por qué recordar lo que hace cada botón, es mejor permitirle reconocerlo mediante ayudas visuales.
- Diseño minimalista, no contener más elementos de los necesarios para la comunicación e interacción.
- Contraste entre los elementos y el fondo para una mejor percepción de estos.
- La organización de los elementos debe ser clara y producir un diseño visualmente satisfactorio.

6.2. Prototipo inicial de la interfaz

Uno de los puntos claves de la metodología iterativa e incremental es trabajar con prototipos de la aplicación en etapas tempranas del proyecto. Mostrar una maqueta del producto nos permite probarlo con los clientes previamente al desarrollo, reduciendo así posibles cambios y correcciones futuras.

A continuación se muestra el prototipo de la aplicación creado mediante la herramienta Figma. En este documento solo se mostrarán las diferentes pantallas con una breve descripción en el pie de las imágenes. Para ver mejor la navegabilidad entre pantallas se puede acceder al proyecto de Figma mediante los enlaces del apéndice D.



Figura 6.1: Vista de inicio de sesión

Nombre único de familia

Nombre del tutor

Correo

Nota: la familia también estará ligada a este correo

Contraseña

Repetir la contraseña

Regístrate



Figura 6.2: Vista de registro en la aplicación

Familia Perez


Padre **Acceder**

Hijo **Acceder**

Madre **Acceder**

Familia Perez

Hijo



Acceder

Familia Perez

Padre

Acceder

Figura 6.3: Vistas de acceso a los perfiles

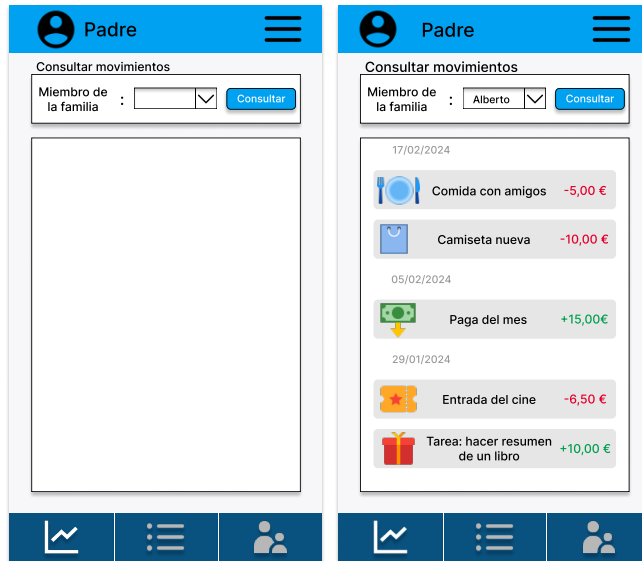


Figura 6.4: Vistas de consulta de movimientos para padres

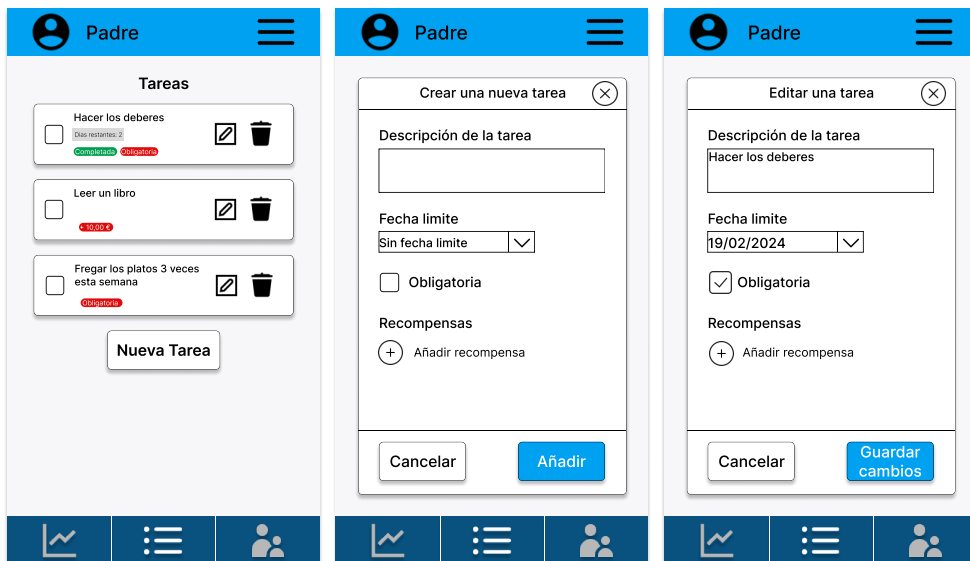


Figura 6.5: Vistas de administración de tareas para padres

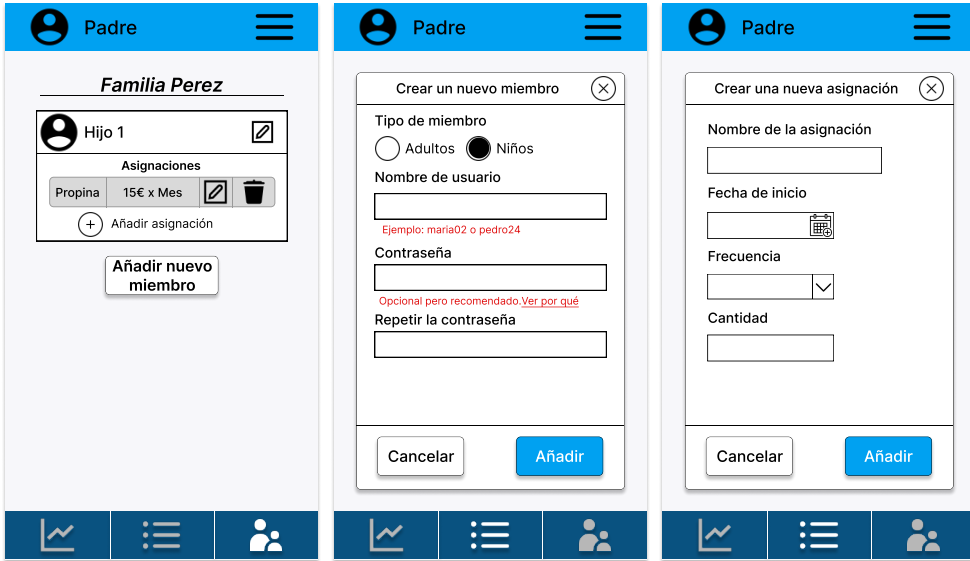


Figura 6.6: Vistas de administración de familia para padres

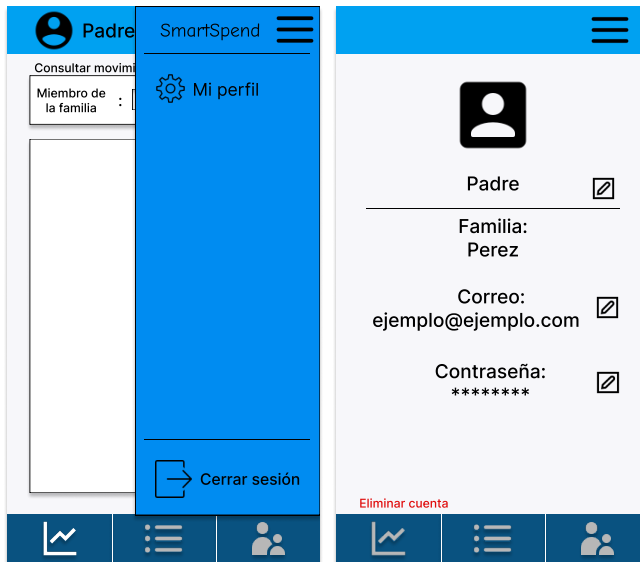


Figura 6.7: Vistas de acceso al perfil

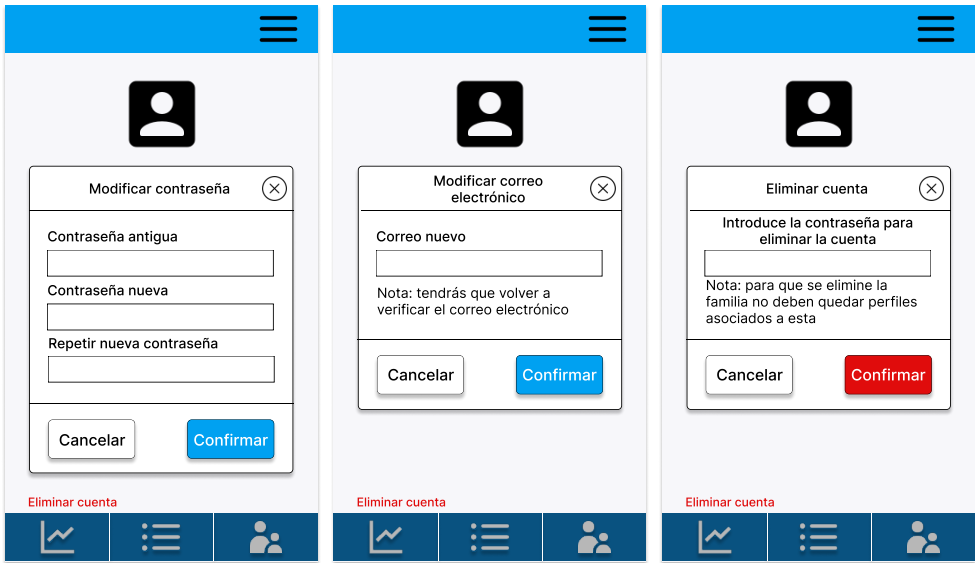


Figura 6.8: Vistas de modificación/eliminación de perfil

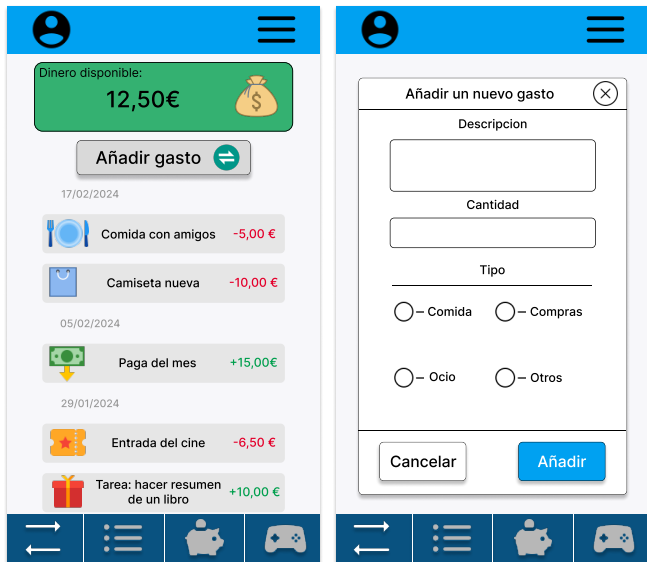


Figura 6.9: Vistas de gestión de movimientos para niños

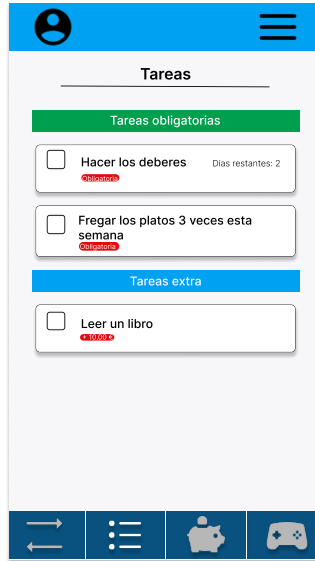


Figura 6.10: Vista de tareas para niños



Figura 6.11: Vistas de objetivos de ahorro para niños



Figura 6.12: Vistas de juegos para niños

6.3. Prueba de satisfacción del usuario

Una vez se ha diseñado un prototipo de la aplicación se debe comprobar si la experiencia de usuario que ofrece es la mejor posible. Para ello, uno de los métodos de prueba más utilizados son las pruebas de usabilidad. Durante estas, los participantes realizan tareas específicas mientras que los evaluadores observan y recopilan datos sobre su desempeño, interacciones y comentarios.

Para este proyecto las pruebas de validación se realizaron cuando la aplicación estaba en un estado avanzado de desarrollo. Esto se debe a una falta de participantes anteriormente y a que el desarrollo de la aplicación no se podía retrasar más para seguir la planificación. Dado que no existe una estructura única para el desarrollo de pruebas de usabilidad, en este caso se siguió la guía proporcionada por un artículo de la revista CONAIC [54].

6.3.1. Planificación de la prueba

- **Objetivos:** el objetivo principal de la prueba es que los usuarios sean capaces de realizar las tareas encomendadas sin ayuda exterior y en un tiempo menor a 10 segundos. Para medir estos parámetros se emplearán las métricas de tasa de éxito dentro del tiempo y tasa de éxito fuera del tiempo. A mayores se tomará nota de los comentarios y observaciones realizados por los participantes.
- **Tareas:** las tareas que se les pedirá a los participantes durante la prueba son:

- **Para los adultos:**

1. Registrarse en la aplicación.
2. Acceder como familia a la aplicación.
3. Seleccionar su perfil.
4. Navegar entre las diferentes funciones principales.
5. Consultar los movimientos de un menor de la familia.
6. Crear una nueva tarea.
7. Revisar y marcar como completada una tarea.
8. Crear un nuevo miembro.
9. Modificar datos de un miembro.
10. Crear una asignación para un menor de la familia.
11. Modificar una asignación.
12. Acceder al menú y al perfil.
13. Modificar datos del perfil.

- **Para los niños:**

1. Acceder al perfil.
2. Navegar entre las diferentes funciones principales.
3. Añadir un gasto.
4. Marcar un tarea como completada.
5. Crear un objetivo.
6. Ingresar y retirar dinero.
7. Acceder a un juego.

- **Planificación de la ejecución:** la prueba de usabilidad consistió en proporcionar un dispositivo móvil, con la aplicación ya instalada a los participantes para posteriormente pedirles realizar las tareas definidas, tomando nota en todo momento de comentarios y resultados.

Resaltar que estas evaluaciones se realizaron en entornos menos formales que los laboratorios, siguiendo lo que se conoce como pruebas de usabilidad de guerrilla o pasillo, es decir, preguntar a conocidos o incluso transeúntes [55].

6.3.2. Desarrollo de la prueba

Finalmente, solo se contó con cuatro participantes del entorno cercano (familiares y compañeros de estudios), lo cual no era el escenario ideal planteado. Sin embargo, se pudo extraer información valiosa de los resultados de la prueba. Estos participantes fueron:

- **Adulto:** un usuario con conocimientos básicos en tecnología y padre de dos niños. Siguió la planificación original y solo completó las tareas específicas de un adulto. Su participación destacó en las partes potencialmente confusas o complejas, ya que el tiempo que tardaba en completar las tareas relacionadas con dichas partes aumentaba significativamente.

- **Menor:** hijo del participante anterior, de 13 años de edad y con conocimientos mínimos sobre tecnología. Aunque está fuera del rango de edad establecido, se adaptaron las tareas dando menos instrucciones y dejando mayor libertad para explorar. Al igual que el anterior participante, solo realizó las tareas específicas de un menor.
- **Expertos en tecnología:** estos usuarios se apartaron del plan original. Al ser expertos en informática realizaron todas las tareas. Su contribución consistió en crear situaciones más allá de lo planeado para identificar posibles errores.

Los resultados recopilados tras la realización de las pruebas se pueden ver en la tabla a continuación 6.2. En esta se anota, empleando números romanos, la cantidad de usuarios que completaron la tarea dentro del límite de tiempo o fuera de él. También se anotaron todos los comentarios realizados durante las pruebas.

Tarea	Tarea completada		Observaciones
	Dentro del tiempo	Fuera del tiempo	
Registrarse en la aplicación.	II	I	Han preguntado: ¿Qué es <i>tutor</i> ?
Acceder como familia a la aplicación.	III		
Seleccionar su perfil.	III		Han preguntado: ¿Cómo se a donde tengo que ir para poner una contraseña?
Navegar entre las diferentes funciones principales.	III		Los usuario tienen preferencia por el uso de botones en lugar de gestos
Consultar los movimientos de un niñ@ de la familia.	III		Han preguntado: ¿Por qué sale una opción en blanco?
Crear una nueva tarea.	III		Han preguntado: Si la tarea es para hoy, ¿por qué sale en rojo los días restantes?
Revisar y marcar como completada una tarea	II	I	
Crear un nuevo miembro.	II	I	
Modificar datos de un miembro.	III		
Crear una asignación para un niñ@ de la familia.	III		
Modificar una asignación.	III		
Acceder al menú y al perfil.		III	Un comportamiento bastante habitual es intentar clicar en el nombre e imagen de perfil
Modificar datos del perfil.	I	II	Han preguntado: ¿Por qué sale una contraseña si el perfil no tiene contraseña?
Acceder al perfil.	II	I	Cuando empieza a aumentar el número de miembros el tiempo que se tarda en realizar la acción también sube.
Navegar entre las diferentes funciones principales.	III		
Añadir un gasto.	II	I	
Marcar un tarea como completada.	III		
Crear un objetivo.	III		
Ingresar y retirar dinero.	I	II	A un usuario el <i>number picker</i> (elemento tipo ruleta como el que hay en muchos teléfonos para seleccionar una hora) le ha resultado un poco incomodo de usar.
Acceder a un juego.	III		Al completar el juego muchos no saben cómo salir del mismo

Como resultado de esta prueba se anotaron una serie de cambios sencillos que mejorarían la usabilidad de la aplicación:

- Modificar el *number picker* para que también se pueda introducir la cantidad mediante teclado, así para cifras grandes no se tardará tanto tiempo en seleccionar.
- Modificar el selector de niños para que no aparezca una opción en blanco.
- Acceder a la pantalla de *mi perfil* al pulsar en el nombre o imagen de usuario en la barra superior.
- En la pantalla de *Mi perfil*, se mostraba siempre un texto con `*****` en el apartado de contraseña. Se cambió para que solo aparezca si el usuario tiene una contraseña configurada para su perfil.
- Para que los usuarios recuerden establecer una contraseña de perfil y, además, sepan donde deben hacerlo, se añade un mensaje en las ventanas emergentes de *perfiles* que no tienen contraseña.
- La pantalla de *registrarse* puede quedarse congelada mientras se crean los datos en la base de datos (2-3 segundos de media), se añade una pantalla de carga para que el usuario sepa que la aplicación esta trabajando en segundo plano.

Y en lo respectivo a errores encontrados durante las pruebas:

- Al crear un usuario no se controlaba el limite de tamaño para el nombre, mientras que al editar los datos del usuario sí.
- El mensaje de error cuando una fecha es anterior a la actual no cambiaba respecto a la de formato incorrecto.
- Se encontró que cuando un menor añadía un gasto y se quedaba con cero dinero en la cuenta, la ventana emergente se cerraba pero no añadía ese gasto ni modificaba el dinero disponible.
- Cuando existía un usuario con imagen de perfil y otro modificaba alguno de sus datos esa imagen se duplicaba en la venta de *perfiles*.
- Error relacionado con los gestos de la aplicación, se mezclaban los gestos de la pantalla con los del menú emergente bloqueando así la interfaz.

6.4. Arquitectura de la aplicación

La aplicación está diseñada siguiendo un paradigma de cliente-servidor, en el cual la aplicación actúa como el cliente y Firebase como el servidor remoto. La aplicación cliente incluye las capas de presentación, que contiene la interfaz de usuario y los elementos con los que el usuario interactúa, y de lógica, encargada de manejar las actividades y la lógica del negocio. Esta sigue una arquitectura MVC (Modelo-Vista-Controlador) que divide las responsabilidades de la aplicación en tres partes [56]:

- **Modelo:** gestiona los datos y la lógica de negocio. Al no estar vinculado a la vista ni al controlador es reutilizable en otros contextos.
- **Vista (XML):** define la representación visual. La vista tiene la responsabilidad de presentar la información al usuario y comunicarse con el controlador cuando el usuario interactúa.
- **Controlador (Activities, fragment y widgets):** maneja la lógica de presentación e interacción, es decir, determina lo que sucede en la aplicación.

En la figura 6.13 se puede observar el comportamiento de la aplicación a nivel general.

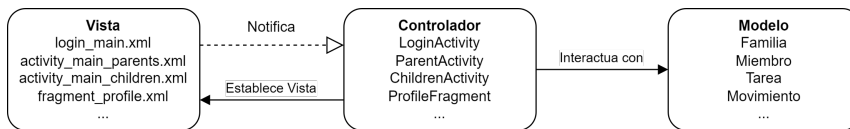


Figura 6.13: Enfoque Modelo-Vista-Controlador

Por otro lado, el *back-end* se basa en Firebase, que se encargará del almacenamiento y manipulación de datos. Firebase simplifica las tareas de configuración del servidor y base de datos, proporcionando un *back-end* listo para usar, también llamado *BaaS* (Back-end as Service) [57]. En este sistema, los datos generados por los usuarios mientras interactúan con la aplicación se almacenan en el *back-end* de Firebase, asegurando que estén sincronizados y actualizados para todos los usuarios.

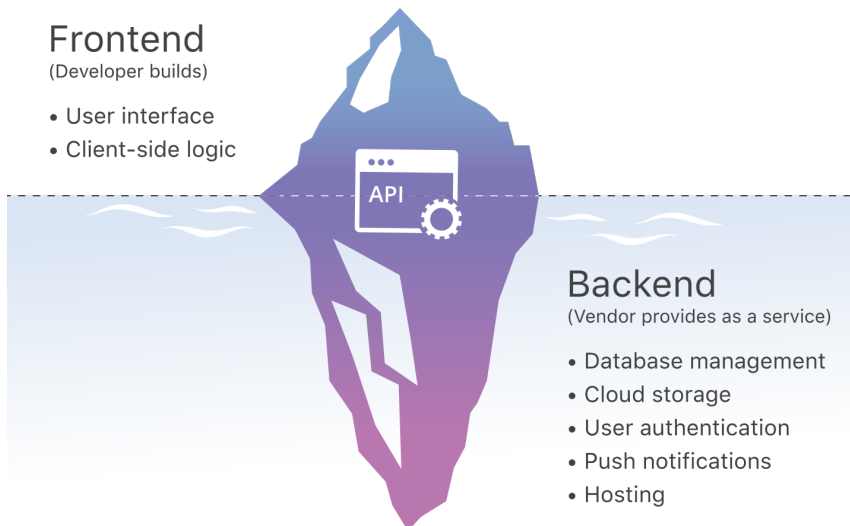


Figura 6.14: Backend-as-a-Service [57]

6.4.1. ¿Por qué un desarrollo nativo en Android?

Emplear un desarrollo nativo tiene múltiples ventajas sobre una arquitectura distribuida (híbrida o web)[58]:

- **Un rendimiento superior:** al estar la aplicación optimizada para el sistema operativo específico (en nuestro caso Android) garantiza un mejor rendimiento que las aplicaciones multiplataforma.
- **Mejora la experiencia del usuario:** al aprovechar al máximo diferentes funciones del sistema operativo, este enfoque proporciona una experiencia más intuitiva y gratificante para el usuario.
- **Acceso a funcionalidades del dispositivo:** las aplicaciones nativas pueden acceder de forma directa a funciones hardware del dispositivo como la cámara.
- **Mejor mantenimiento y soporte:** al centrarse en una única plataforma es más fácil proporcionar actualizaciones y mejoras de seguridad que desarrollando para múltiples plataformas.

A cambio perdemos la compatibilidad de una aplicación distribuida, con lo que llega a un menor número de usuarios, sin embargo, es un coste asumible debido a que nuestro público, en su mayoría niños, suelen utilizar dispositivos móviles como principal medio.

En resumen, queremos proporcionar la mejor experiencia de usuario posible y ahí es donde entra el rendimiento de la aplicación. En un trabajo de fin de grado de la universidad de Ocaña (Colombia) se comparó el rendimiento de una misma aplicación desarrollada mediante los dos enfoques, demostrando que la versión nativa es más rápida y consume muchos menos recursos que su versión basada en web [59].

6.5. Back-end

Como se menciona en la arquitectura, gran parte de las funciones *back-end* de una aplicación recaerá sobre Firebase. En la subsección 4.4 se explica qué es Firebase y los servicios que ofrece. Dichos servicios están alojados en la nube, y Google se encarga de gestionarlos y mantenerlos. De esta forma las aplicaciones cliente pueden emplear estos servicios mediante la invocación de *end-points*, lo cual reduce mucho los tiempos de desarrollo, sobretodo si el *back-end* consiste únicamente en operaciones CRUD y de registro de usuarios. Esto lo convierte en una buena opción para nuestra aplicación, donde nuestro sistema se encargará de toda la lógica e interfaz, relegando las operaciones de intercambio de datos a Firebase.

De entre todos los servicios que ofrece Firebase se emplean los siguientes:

- **Firestore Authentication:** para el registro e identificación de los usuarios.

- **Cloud Firestore:** una base de datos noSQL, rápida y en la nube que permite mantener sincronizados los datos de todos los usuarios en múltiples dispositivos.
- **Firebase Storage:** almacena contenido creado por el usuario, en nuestro caso serán sus imágenes de perfil.
- **App Check:** proporciona una capa de seguridad extra ante ataques o uso indebido de nuestras APIs.

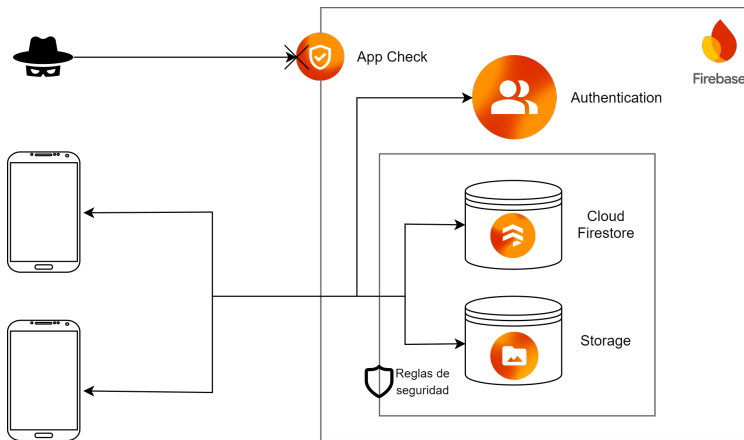


Figura 6.15: Estructura del back-end

Respecto a la base de datos, esta es NoSQL, lo cual tiene sus ventajas e inconvenientes [60]:

Ventajas:

- **Fácil escalabilidad:** está diseñado para escalar horizontalmente distribuyendo la carga a través de múltiples servidores.
- **Flexibilidad:** permite modelos de datos flexibles y sin esquemas fijos, además de soportar diferentes modelos de datos.
- **Alto rendimiento y baja latencia:** maneja grandes cantidades de datos a alta velocidad, lo cual es especialmente útil en aplicaciones en tiempo real.

Desventajas:

- **Menor consistencia:** se sacrifica la consistencia que proporcionan los modelos SQL a favor de una menor latencia y la distribución de los datos.
- **Consultas limitadas:** a menudo las consultas se limitan a simples operaciones de lectura y escritura, quedando en manos de la aplicación la gestión y filtrado de los datos.
- **Falta de estandarización:** no existe un estándar global y bien establecido sobre las bases de datos NoSQL y la migración de datos puede ser más difícil debido a la falta de estos estándares.

Toda esta información se tuvo en cuenta y se llegó a la decisión de que al trabajar con una cantidad de datos relativamente pequeña podríamos sacar el máximo partido de una base de datos NoSQL.

6.5.1. Estructura de la base de datos

Cloud Firestore está organizado en colecciones y documentos. Las colecciones actúan como los contenedores de varios documentos. Estos a su vez, poseen campos donde se almacenan diversos tipos de datos (números, cadenas de texto, marcas de tiempo, etc.). Además, es posible crear subcolecciones dentro de un documento, permitiendo estructurar los datos de forma jerárquica.

En la figura a continuación 6.16 se puede observar la estructura de la base de datos del proyecto, representando las colecciones con la abreviatura *col* y los documentos con *doc*. Justo a continuación se detallan los datos almacenados en cada tipo de documento.

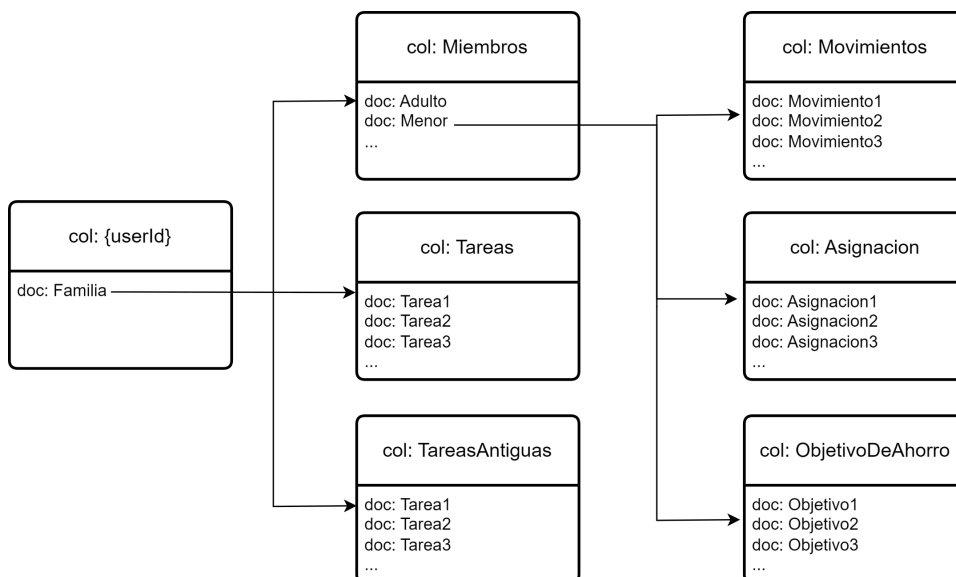


Figura 6.16: Estructura de la base de datos

Familia

- **emailFamilia:** String
- **nombreFamilia:** String

Adulto

- **usuario:** String
- **contraseña:** String
- **tipo:** String

Menor

- **usuario:** String
- **contraseña:** String
- **tipo:** String
- **dinero:** String

Tareas

- **descripción:** String
- **obligatoria:** Boolean
- **fechaLimite:** String
- **recompensa:** String
- **estado:** String
- **fechaCompletada:** String
- **menor:** String
- **asignada:** Boolean

Movimientos

- **descripción:** String
- **cantidad:** String
- **fecha:** String

- **tipo:** String

Asignación

- **nombre:** String
- **cantidad:** String
- **proximoPago:** String
- **tipo:** String
- **caducada:** Boolean

Objetivo de ahorro

- **descripción:** String
- **tipo:** String
- **objetivo:** String
- **ahorrado:** String
- **conseguido:** Boolean

A continuación se explican algunas decisiones de diseño:

- El dinero se almacena en una cadena de caracteres en vez de en un tipo numérico debido al uso de *BigDecimal* dentro de la aplicación. De esta forma evitamos inexactitudes en los decimales por el uso del punto flotante y las conversiones de tipos.
- Si una tarea está asignada a un menor específico, el campo *menor* siempre contendrá el ID de ese menor. Si la tarea no está asignada, ese campo permanecerá vacío hasta que alguien marque la tarea como completada, momento en el cual se guardará el ID de quien la completó. El campo *fechaCompletada* se rellenará cuando un adulto revise y valide la tarea.
- El campo *Tipo* en *Adulto* y *Menor* sirven para diferenciar los tipos de usuario dentro de la aplicación ya que para Firebase estos no son más que documentos y no tienen diferencias entre ellos.

6.5.2. Reglas de seguridad

Aunque App Check nos proporciona una capa de seguridad ante ataques externos a nuestra aplicación debemos evitar que algún usuario pueda alterar los datos de otro. Esto se consigue mediante la definición de lo que en Firebase se llaman reglas. Dada la estructura de la base de datos, en la que todos los datos de cada familia se encuentran en una única colección, solo es necesario definir reglas que protejan esa colección durante las operaciones de lectura y escritura.

De modo que las reglas para Cloud Firestore son de la siguiente manera:

```
1 service cloud.firestore {
2   match /databases/{database}/documents {
3     match /{userId}/{documents=**} {
4       allow read, update, delete: if request.auth != null && request.
5         auth.uid == userId;
6       allow create: if request.auth != null;
7     }
8     match /mock/{documents=**} {
9       allow read: if true;
10    }
11  }
```

Listing 6.1: Reglas de Cloud Firestore

Y las de Firebase Storage:

```
1 service firebase.storage {
2   match /b/{bucket}/o {
3     match /images/{userId}/{documents=**} {
4       allow read, update, delete: if request.auth != null && request.
5         auth.uid == userId;
6       allow create: if request.auth != null && request.auth.uid ==
7         userId && request.resource.contentType.matches('image/.*');
8     }
9   }
10 }
```

Listing 6.2: Reglas de Firebase Storage

De esta forma solo un usuario registrado (o familia en nuestro caso) puede crear una nueva colección y posteriormente solo él puede editarla, leerla o eliminarla. Se añade una excepción para la lectura de la colección *mock*, que sigue la misma estructura solo que con datos predefinidos para que los usuarios puedan probar la aplicación.

6.6. Front-end

Tal y como se menciona en el apartado sobre la arquitectura de la aplicación 6.4, el *front-end* es una aplicación Android programada en Kotlin. Los usuarios interactúan con la aplicación y esta se comunica con el *back-end* en segundo plano, para que los datos estén siempre sincronizados con la nube.

6.6.1. Patrones de diseño

Un patrón de diseño es una solución reusable a un problema común en el diseño de software. En esta subsección se detallan los patrones de diseño empleados en la creación de la aplicación.

Patrón Adapter

Un adaptador es un objeto especial que convierte la interfaz de un objeto, de manera que otro objeto pueda comprenderla. Permite que objetos incompatibles trabajen juntos al envolver la interfaz de uno de ellos para que sea compatible con el otro [61].

En el contexto de la aplicación los adaptadores se encargan de encapsular los objetos del modelo para representarlos en la interfaz de usuario. Por ejemplo, en un RecyclerView, el adaptador toma datos del modelo y los convierte en vistas que pueden ser mostradas en la interfaz (puede verse con más detalle en la figura 6.17).

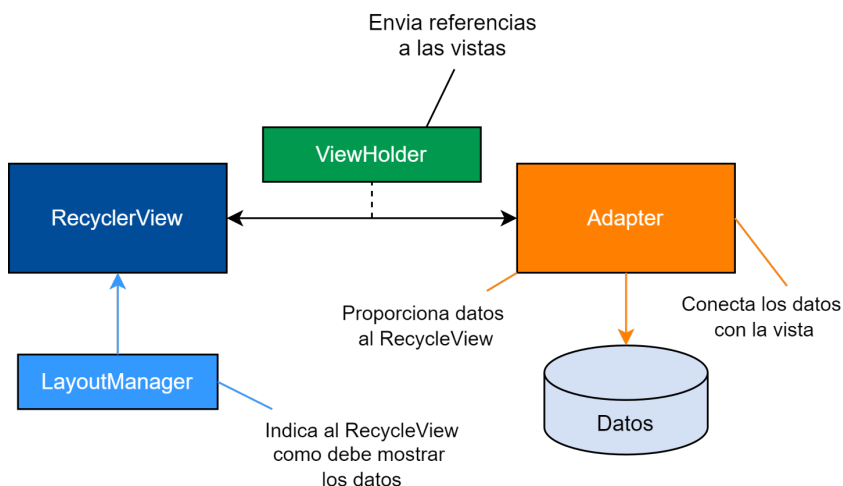


Figura 6.17: Funcionamiento del RecyclerView [62]

Patrón Singleton

Un *singleton* [63] es un patrón de diseño que restringe la instanciación de una clase a un solo objeto. Kotlin facilita la creación de *singletons* a través de la palabra clave *object*. Para nuestro caso, necesitamos un gestor para la familia identificada en la aplicación. Este gestor debe garantizar que solo haya una instancia de la clase *Familia* y proporcionar un acceso global a dicha instancia. Existen varias razones para ello:

1. Garantizar que en toda la aplicación haya solo una instancia de la clase *Familia*
2. Optimiza el uso de memoria al evitar la repetición de datos entre ventanas.
3. Mantener el estado de la familia, lo cual es crucial para garantizar coherencia de los datos y prevenir problemas de sincronización.
4. Proporciona un método simple y directo para acceder a la información de la familia sin necesidad de crear instancias adicionales.

De forma similar los servicios de Firebase siguen un patrón *singleton*. Cuando se instancia un servicio de Firebase, como `FirebaseAuth` se obtiene una instancia única de ese servicio. Por ejemplo, al escribir `FirebaseAuth.getInstance()`, se obtiene una referencia a la única instancia de `FirebaseAuth` en la aplicación. Cualquier otra llamada a `FirebaseAuth.getInstance()` en partes diferentes de la aplicación devuelve esa misma instancia.

6.6.2. Estructura de la aplicación

A continuación se detalla la estructura que sigue la aplicación (figuras 6.18, 6.19, 6.20, 6.21):

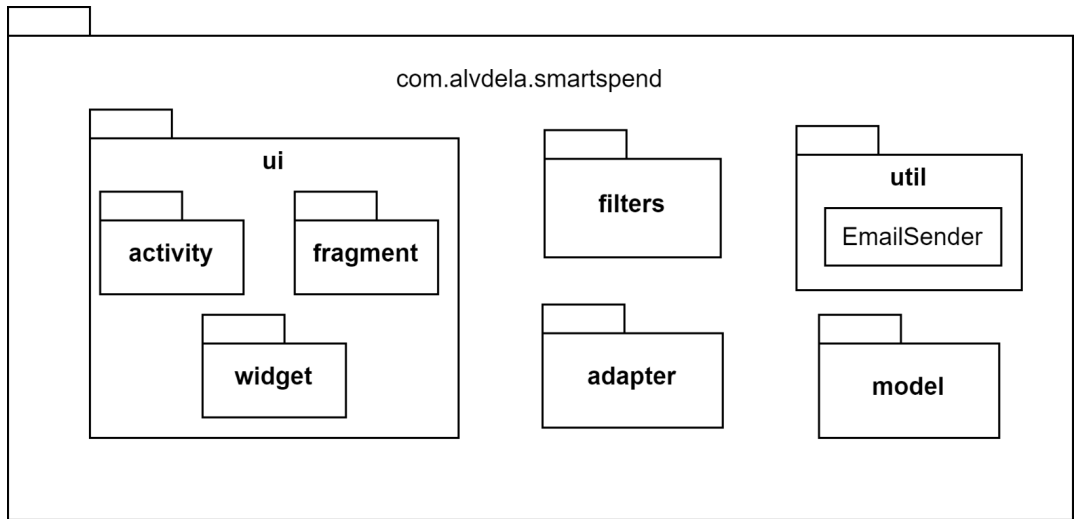


Figura 6.18: Diagrama de paquetes de la aplicación Android

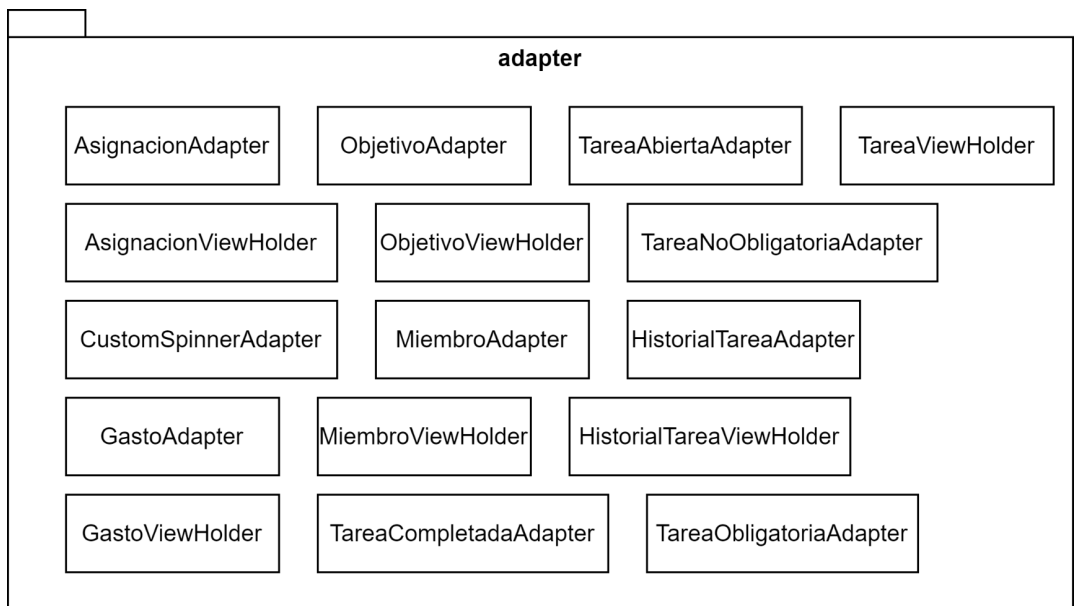


Figura 6.19: Paquete adapter, almacena las clases encargadas de transformar un modelo de datos en un elemento visual

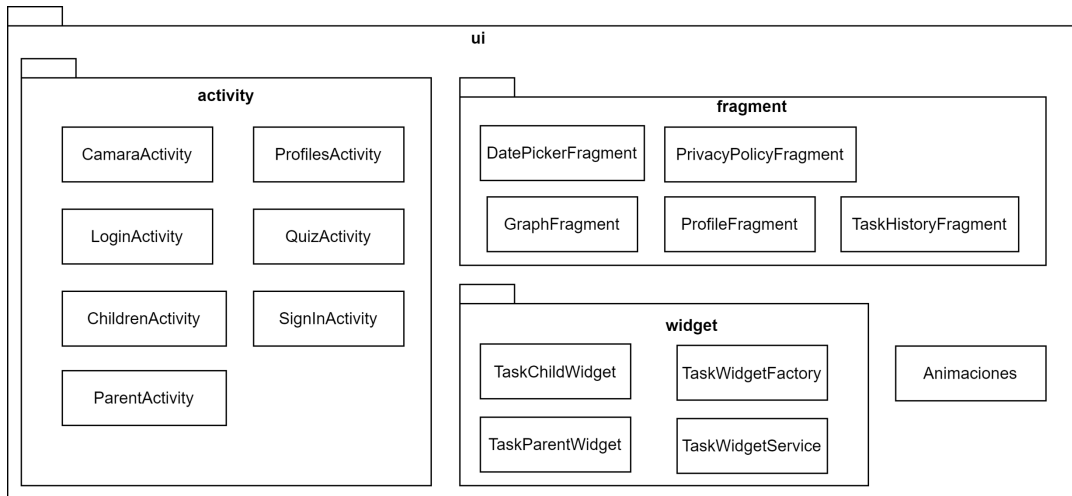


Figura 6.20: Paquete ui, almacena las clases Controlador (Activities, fragments y widgets)

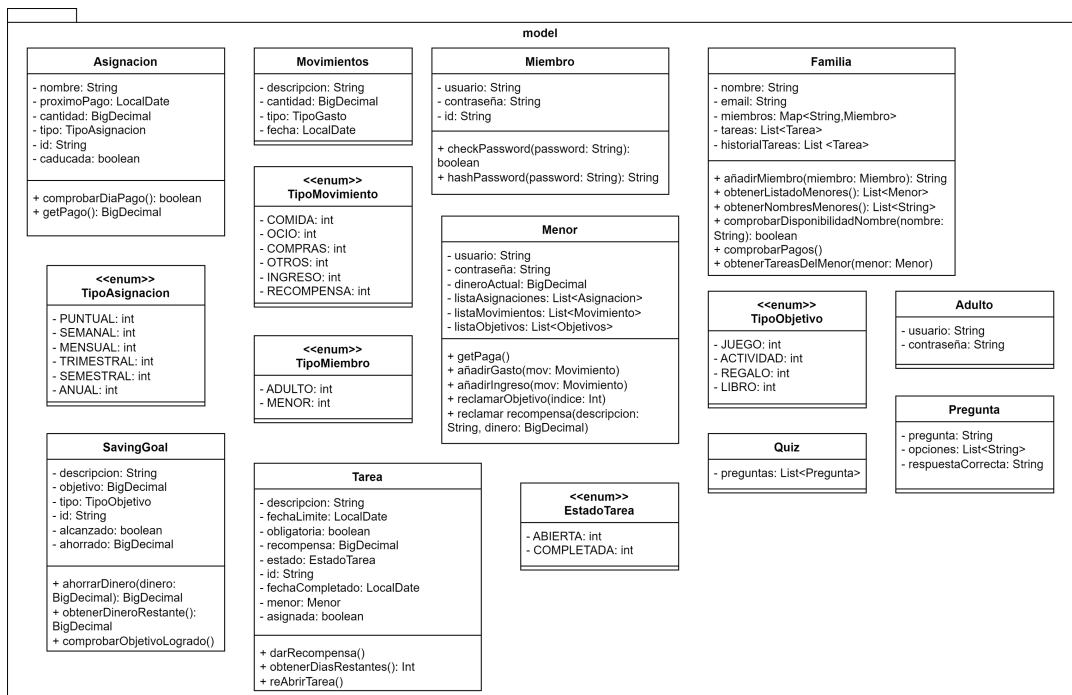


Figura 6.21: Paquete model, almacena las clases del modelo de dominio (se omiten las funciones getter y setter)

Capítulo 7

Implementación y pruebas

7.1. Formación previa

Con el objetivo de mitigar el riesgo RSK003 3.4 (Falta de formación en la tecnología) el alumno realizó una formación online previa al desarrollo de la aplicación. Esta formación comenzó con una introducción a los conceptos principales del lenguaje Kotlin (variables, funciones, clases, etc.) para posteriormente meterse de lleno en los conceptos del desarrollo de aplicaciones. Se explica cómo se estructura un proyecto Android, el comportamiento de los Activities, creación de layouts y views personalizados, entre otras muchas cosas que han sido de utilidad para el proyecto.

Puede verse parte del código desarrollado durante la formación en los repositorios de Github desde el apéndice D.

7.2. Iteraciones de desarrollo

Como se ha detallado en el capítulo 3, la metodología iterativa se basa en diferentes iteraciones de desarrollo en las cuales se van añadiendo diferentes funcionalidades y mejoras respecto a la versión anterior. Cada iteración tendrá una planificación previa al desarrollo y una evaluación de los resultados al finalizar.

7.2.1. Primera iteración (12/3/2024 - 9/4/2024)

Objetivo de la iteración

Crear las pantallas principales e implementar las funcionalidades de añadir gastos, todo ello mediante la opción de acceso como invitado, prescindiendo de una base de datos.

Tareas

- Tarea 1: crear pantallas principales y su navegación entre ellas.
- Tarea 2: implementar el acceso mediante el modo invitado.
- Tarea 3: implementar la funcionalidad de añadir gastos en el menú de los niños.
- Tarea 4: implementar la funcionalidad de ver los gastos de los hijos en el menú de los padres.

Resultados de la iteración

Se consiguió completar los objetivos planteados para la iteración y se realizaron las siguientes observaciones durante la reunión de seguimiento:

- Puede ser poco intuitivo que existan movimientos fuera de la parte visible de la pantalla y que son accesibles mediante la acción de *scroll*. Posibles soluciones para ello: contraste de colores y añadir barra de *scroll* vertical.
- Arreglar error con el filtro de introducir cantidad de dinero de un nuevo gasto, no deja borrar el primer número una vez está escrito.
- Mejorar los menús emergente o *pop up* para que no permitan clicar en otros elementos de la pantalla hasta que estos se cierren.

Riesgos identificados

Durante esta iteración surgieron los riesgos RSK003 (3.4), debido a la falta de experiencia trabajando con algunos elementos de *Android*, y el RSK008 (3.9), debido a que desde enero el editor *LaTeX Overleaf* limitó los tiempos de compilación a menos que se pague una suscripción.

7.2.2. Segunda iteración (9/4/2024 - 22/4/2024)

Objetivo de la iteración

Solucionar los errores, hacer las modificaciones mencionadas durante la iteración anterior y añadir la función de administración en el perfil de los adultos.

Tareas

- Tarea 1: arreglar el filtro de la cantidad de dinero.
- Tarea 2: mejorar los menús emergentes o *pop ups*.
- Tarea 3: mejoras en la interfaz y auto completado en la descripción de los gastos.

- Tarea 4: implementar la funcionalidad de administración de perfiles.
- Tarea 5: implementar la funcionalidad de asignaciones o propinas.

Resultados de la iteración

Los resultados de esta interacción no fueron como se estimaron y no se pudo completar todas las tareas planificadas, por lo que la tarea número 5 se pasó a la siguiente iteración.

En lo respectivo a la reunión de seguimiento se realizaron pocos comentarios en lo relacionado a la implementación y más en lo respectivo a la parte de la memoria. Durante esta reunión también se comentó cambiar los nombres de los tipos de usuarios. De esta forma *padres* pasaría a ser *adultos* y *niños* pasaría a ser *menores*. Estos nombres son menos restrictivos ya que pueden existir casos de tíos y sobrinos, hermanos mayores y pequeños, etc.

Riesgos identificados

En esta iteración surgió un nuevo riesgo 3.17 relacionado con la preparación de exámenes, el cual está analizado en la sección 3.8. Junto con este riesgo también se materializó el 3.5, cuya acción correctiva fue incrementar el número de horas dedicadas al proyecto.

7.2.3. Tercera iteración (22/04/2024 - 06/05/2024)

Objetivo de la iteración

Completar la funcionalidad de propinas, desarrollar las funcionalidades de tareas y objetivos de ahorro y comenzar a integrar el *back-end* de *Firebase* (*Auth* y *Cloud Firestore*).

Tareas

- Tarea 1: completar el desarrollo de la funcionalidad de propinas.
- Tarea 2: desarrollar la funcionalidad de tareas en el perfil de los adultos.
- Tarea 3: desarrollar la funcionalidad de tareas en el perfil de los menores.
- Tarea 4: implementar la funcionalidad de objetivos de ahorro para los menores.
- Tarea 5: comenzar la integración de servicios de *Firebase*.

Resultados de la iteración

Los resultados de esta iteración fueron positivos y se completaron todas las tareas planificadas para la iteración. Además de dichas tareas se avanzó con la función de *mi perfil*, se hizo una refactorización y optimización del código y se añadieron mejoras visuales a nivel de interfaz como un *SplashScreen* (ventana que aparece al iniciar la aplicación) o una ventana que muestra visualmente la descarga de datos desde la nube. Durante la reunión de seguimiento se hicieron las siguientes observaciones:

- Algunos usuarios pueden usar la misma contraseña tanto para la cuenta de la familia como para su perfil. Guardar la contraseña sin cifrar en la base de datos puede ser un fallo de seguridad grave.
- Durante la demo fueron apareciendo diferentes errores de los que se tomó nota para eliminarlos en la siguiente iteración.

Riesgos identificados

El único riesgo identificado durante esta iteración fue el *gold plating* 3.8, debido a que se desarrollaron elementos no contemplados en la planificación de la iteración, sobre todo a nivel de interfaz. Al no tener un impacto alto para el proyecto, simplemente se ignorará y no se tomarán ninguna acción correctiva al respecto.

7.2.4. Cuarta iteración (06/05/2024 - 14/05/2024)

Objetivo de la iteración

Solucionar errores, cifrar las contraseñas para su almacenamiento, desarrollar las funciones de *mi perfil*, *historial de tareas* y pulir la integración con la base de datos *Cloud Firestore*.

Tareas

- Tarea 1: solucionar errores (visual reabrir tarea y conseguir objetivo).
- Tarea 2: cifrar contraseña.
- Tarea 3: desarrollar las funciones de mi perfil, historial tareas y poder añadir fotos de perfil.
- Tarea 4: terminar base de datos.
- Tarea 5: permitir el borrado de datos de los usuarios.

Resultados de la iteración

Durante esta iteración fueron apareciendo más errores de los previstos y se dio prioridad a la depuración de estos antes de añadir nuevas funcionalidades (riesgo 3.18). Además de los errores visuales, causados por cómo se actualizaba la interfaz, existía uno que afectaba a todo el funcionamiento de la aplicación. Este error era producido por la inexactitud respecto a los decimales al trabajar con punto flotante, por lo que se hizo un cambio de todas las variables que empleaban esta unidad por la clase *BigDecimal*. Las tareas 3 y 5 quedaron incompletas y se trasladaron a la siguiente iteración.

Durante la reunión de seguimiento aparecieron nuevos errores a tratar entre los cuales se puede destacar:

- Para acceder a un perfil en el que se olvidó la contraseña se accede mediante la contraseña de la familia, si esa contraseña se modifica ya no se puede acceder por este método.

Riesgos identificados

El riesgo principal es, una vez más, no cumplir con la planificación inicial 3.4, para el cual se tomó de nuevo la acción de incrementar el numero de horas dedicadas. A este riesgo hay que añadir una mala estimación de la duración de las tareas 3.18 y la aparición de nuevos errores que corregir 3.19.

7.2.5. Quinta iteración (14/05/2024 - 23/05/2024)

Objetivo de la iteración

Completar todas las tareas de la iteración anterior y la creación de un pequeño juego educativo.

Tareas

- Tarea 1: solución de errores.
- Tarea 2: solucionar problema con el acceso al perfil mediante la contraseña de familia.
- Tarea 3: terminar el desarrollo de mi perfil, historial tareas y función de añadir fotos.
- Tarea 4: implementar el borrado de datos de la familia.
- Tarea 5: crear un pequeño juego educativo (tipo cuestionario).

Resultados de la iteración

Los resultados de esta iteración fueron positivos quedando únicamente el juego sin terminar de desarrollar, pero en un estado avanzado. Durante la reunión se comentaron los siguientes puntos:

- El envío de un correo electrónico para confirmar el cambio de correo electrónico de la cuenta a veces falla.
- Cuando se elimina una cuenta por completo se produce un error si queremos acceder al modo invitado.

Riesgos identificados

Los principales riesgos durante esta iteración son: una mala estimación del tiempo necesario para desarrollar el juego 3.18 y la aparición de errores 3.19. Sin embargo, estos riesgos no afectaron a la planificación general del proyecto, estimando que con la siguiente iteración el proyecto podría quedar completado.

7.2.6. Sexta iteración (23/05/2024 - 30/05/2024)

Objetivo de la iteración

El objetivo de esta iteración es rematar las últimas tareas de desarrollo y depurar la aplicación de errores para tener el mejor producto final posible. A este objetivo se añade completar la memoria del proyecto.

Tareas

- Tarea 1: arreglar que se envíe el correo de confirmación de cambio de correo electrónico.
- Tarea 2: solucionar algunos errores.
- Tarea 3: completar el juego.
- Tarea 4: nueva pestaña para padres que muestra gráficos de como han gastado el dinero sus hijos.
- Tarea 5: enviar notificaciones y que el padre pueda configurar recibirlas o no.
- Tarea 6: rematar la memoria del proyecto.

Resultados de la iteración

Tras esta iteración la parte de implementación queda finalizada, por lo que será la última iteración documentada. El trabajo restante consistió en realizar las pruebas de aceptación, mantenimiento de la aplicación y finalización de la memoria.

Riesgos identificados

En esta iteración los principales riesgos fueron la aparición de errores 3.19 y el *gold plating* 3.8. Durante esta iteración este segundo riesgo se ha notado más debido a que se crearon *widjets* para las tareas y se implementó la navegación por gestos, todo ello sin estar en la planificación inicial.

7.3. Organización del código

El flujo de trabajo de Git seguido para este proyecto es la conocida como GitFlow, con pequeñas modificaciones. Este flujo de control del código consiste principalmente en tener dos ramas en lugar de una: *main* y *develop*. La rama *main* almacena únicamente las publicaciones oficiales mientras que *develop* sirve como rama de integración y desarrollo de las funciones [64].

Cuando se ha desarrollado lo suficiente la aplicación para publicarla, se debe bifurcar a una rama *release* a partir de *develop*. Durante esta fase no se deben añadir nuevas funcionalidades,

únicamente arreglar errores y generar la documentación. Una vez que la aplicación está realmente lista para su liberación, *release* se unifica con *main* y se le asigna un numero de versión (también se une *release* con *develop* para que el desarrollo este actualizado con los cambios).

En este TFG no se han tenido en cuenta las ramas *feature* y *hotfix* propuestas en la metodología GitFlow, ya que podrían resultar en incrementar los tiempos de desarrollo y añadir complejidad innecesaria a la organización del código, por lo que todos los desarrollos y cambios se hacen en la rama *develop*.

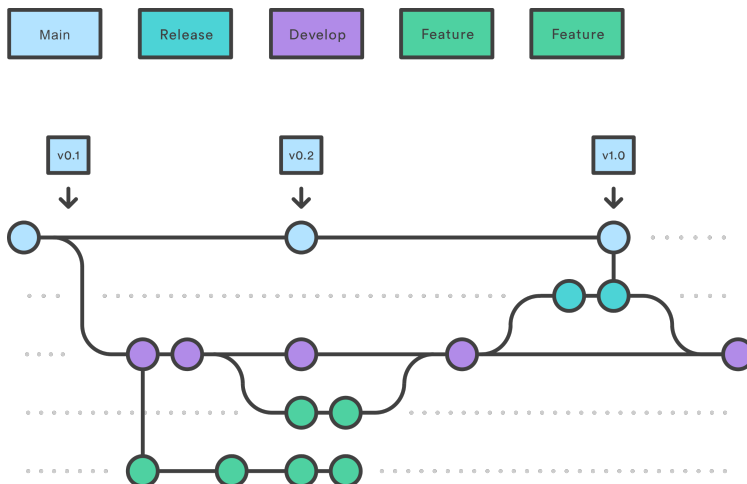


Figura 7.1: Ejemplo visual de GitFlow[64]

7.4. Dificultades y retos

De todas las tareas que se han realizado a lo largo del proyecto algunas han supuesto un reto mayor que las demás. A continuación se citan algunas de las que se han percibido más difíciles:

- **Modificación de datos:** inicialmente la base de datos estaba diseñada de forma que la familia se guardaba mediante el correo electrónico y los usuarios por sus nombres. Esto supuso un problema al añadir la función de cambio de correo y de nombre de usuario, pues Firebase no deja cambiar el nombre de sus colecciones y documentos una vez creados. Ante esto solo existían dos soluciones: cuando se haga una modificación trasladar toda la información almacenada bajo esa colección o documento a uno nuevo, lo cual era sumamente costoso, o cambiar esos campos por un id. Se tomó esta segunda decisión y se tuvo que refactorizar todo el código relacionado.
- **Imágenes de perfil:** la dificultad de este reto surge de la inexperiencia al trabajar con hardware del dispositivo desde código. Se tuvieron que hacer muchas pruebas hasta entender el funcionamiento por completo y refinar el resultado.

- **Widgets:** la parte de *widgets* también fue complicada debido a que nunca se había desarrollado este tipo de componente Android. Estos funcionan como elementos externos a la aplicación y se actualizan como tales. Esto añadido al hecho de que no aceptan contenedores como los *RecyclerView*, que simplifican la generación de elementos visuales en base a datos del modelo, complicó todo su desarrollo.

7.5. Política de privacidad

Dado que nuestra aplicación está dirigida tanto a adultos como a menores, es necesario establecer una política de privacidad que garantice la seguridad de estos y de sus datos. Por ello, se ha dedicado tiempo a crear una política adaptada a la aplicación y a los servicios que ofrece. Para desarrollarla, el alumno se basó en una plantilla y luego realizó las modificaciones necesarias para que se ajustara adecuadamente a la aplicación.

Esta política se puede consultar con más detalle dentro del apéndice C.

7.6. Seguridad de la aplicación

Para garantizar la seguridad de los datos, sobre todo de los menores, se han implementado varias medidas de seguridad. En primer lugar, todo el *back-end* está protegido contra ataques externos mediante *App Check* de Firebase, el cual bloquea todas las peticiones a menos que se envíe un token generado por nuestra aplicación. Además, los datos de cada familia están seguros frente a otros usuarios de la aplicación gracias a las reglas establecidas de Firebase, que solo permiten leer y modificar los datos de la propia familia.

Finalmente, la única información sensible son las contraseñas de los perfiles, debido a que los usuarios pueden utilizar la misma contraseña en varias cuentas. Estas contraseñas se encriptan tanto localmente como en la nube empleando un código *hash*. El resto de la información, tanto de menores como de adultos, no es sensible, ya que no se solicita ninguna información delicada como DNI, apellidos, números de teléfono, etc. y el dinero solo funciona como una emulación, no como dinero real.

7.7. Licencia

Licencia Creative Commons

La aplicación SmartSpend está bajo una licencia Creative Commons de tipo Atribución-NoComercial (BY-NC). Esta licencia permite a otros compartir y adaptar esta obra de forma no comercial, siempre y cuando se dé crédito adecuadamente al autor original.

Para más detalles sobre esta licencia, visite el siguiente enlace: [Creative Commons BY-NC 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/).

7.8. Pruebas y evaluación de la app

En esta sección veremos las pruebas de aceptación llevadas a cabo. Estas pruebas de aceptación tienen como objetivo verificar que el software cumple con los requisitos y que está listo para ser liberado.

7.8.1. Casos de pruebas

Registrarse	CP01
Descripción: Un usuario se registra en la aplicación.	
Pasos: El usuario accede al apartado registrarse, rellena todos los datos, acepta la política de privacidad y pulsa el botón <i>Registrarse</i> .	
Resultado esperado: Se registra el usuario en Firebase Auth, se crea el documento de familia correspondiente en la base de datos, y regresa al inicio.	
Resultado obtenido: Correcto.	

Tabla 7.1: CP01. Registrarse.

Registrarse (faltan datos)	CP01-1
Descripción: Un usuario se intenta registrar pero sin completar el formulario.	
Pasos: El usuario accede al apartado registrarse y pulsa el botón <i>Registrarse</i> sin completar la información.	
Resultado esperado: Se comunica al usuario de los datos que faltan.	
Resultado obtenido: Correcto.	

Tabla 7.2: CP01-1. Registrarse(faltan datos).

Registrarse (correo no válido)	CP01-2
Descripción: Un usuario se intenta registrar con un correo no valido.	
Pasos: El usuario intenta registrarse mediante un correo con un formato incorrecto o que ya está ocupado.	
Resultado esperado: Se informa al usuario de que el correo no es válido y por qué.	
Resultado obtenido: Correcto.	

Tabla 7.3: CP01-2. Registrarse(correo no válido).

Registrarse (Contraseña insuficiente)	CP01-3
Descripción: Un usuario se intenta registrar con una contraseña insuficiente.	
Pasos: El usuario intenta registrarse con una contraseña sin seguridad mínima.	
Resultado esperado: Se informa al usuario de que la contraseña es insuficiente.	
Resultado obtenido: Correcto.	

Tabla 7.4: CP01-3. Registrarse (Contraseña insuficiente).

Iniciar sesión	CP02
Descripción: El usuario inicia sesión.	
Pasos: El usuario introduce correo y contraseña, y pulsa el botón <i>Acceder</i> .	
Resultado esperado: Su guarda la sesión y se muestran los perfiles de la familia.	
Resultado obtenido: Correcto.	

Tabla 7.5: CP02. Iniciar sesión.

Iniciar sesión (Datos incorrectos)	CP02-1
Descripción: Un usuario intenta iniciar sesión con el correo y/o contraseña erróneos.	
Pasos: El usuario escribe su correo o contraseña de forma incorrecta y pulsa el botón <i>Acceder</i> .	
Resultado esperado: Se informa al usuario de que el correo o la contraseña es incorrecta.	
Resultado obtenido: Correcto.	

Tabla 7.6: CP02-1. Iniciar sesión (Datos incorrectos).

Iniciar sesión (Correo no confirmado)	CP02-2
Descripción: Un usuario intenta iniciar sesión sin confirmar su correo.	
Pasos: El usuario escribe su correo y contraseña y pulsa el botón <i>Acceder</i> .	
Resultado esperado: Se informa al usuario de que debe confirmar su correo electrónico.	
Resultado obtenido: Correcto.	

Tabla 7.7: CP02-2. Iniciar sesión (Correo no confirmado).

Acceder a un perfil	CP03
Descripción: Un usuario accede a su perfil.	
Pasos: El usuario pulsa en su perfil, introduce su contraseña (si tiene) y pulsa el botón <i>Acceder</i> .	
Resultado esperado: Se muestra la pantalla principal para el tipo de usuario correspondiente.	
Resultado obtenido: Correcto.	

Tabla 7.8: CP03. Acceder al perfil de adulto.

Inicio de sesión automático	CP04
Descripción: La aplicación accede directamente al apartado <i>perfiles</i> .	
Pasos: Se ha cerrado la aplicación sin cerrar sesión.	
Resultado esperado: Se accede directamente al apartado <i>perfiles</i> , sin necesidad de identificarse de nuevo.	
Resultado obtenido: Correcto.	

Tabla 7.9: CP04. Inicio de sesión automático.

Crear un nuevo usuario	CP05
Descripción: Un adulto crea un nuevo perfil de usuario.	
Pasos: El adulto pulsa el botón <i>Añadir miembro</i> , elige el tipo de usuario e introduce un nombre como mínimo.	
Resultado esperado: Se crea el documento correspondiente en la base datos y se actualizan todas las vistas relacionadas.	
Resultado obtenido: Correcto.	

Tabla 7.10: CP05. Crear un nuevo usuario.

Modificar los datos de un usuario	CP06
Descripción: Un adulto modifica los datos de otro usuario.	
Pasos: El adulto pulsa el botón con un lápiz en el perfil objetivo e introduce los nuevos datos.	
Resultado esperado: Se modifican los datos en el documento de la base datos y se actualizan todas las vistas relacionadas.	
Resultado obtenido: Correcto.	

Tabla 7.11: CP06. Modificar los datos de un usuario.

Restablecer contraseña del perfil	CP07
Descripción: Un usuario quiere restablecer la contraseña de su perfil.	
Pasos: Pulsa el botón <i>He olvidado la contraseña</i> en la ventana emergente de acceder al perfil e introduce la contraseña de la cuenta familiar.	
Resultado esperado: Se elimina la contraseña de ese perfil y accede a la ventana correspondiente .	
Resultado obtenido: Correcto.	

Tabla 7.12: CP07. Restablecer contraseña del perfil.

Crear una asignación	CP08
Descripción: Un adulto crea una asignación para un menor de la familia.	
Pasos: El adulto pulsa el botón <i>Añadir asignación</i> , rellena los campos y pulsa <i>añadir</i> .	
Resultado esperado: Se crea la asignación dentro del documento del menor y se actualizan las vistas.	
Resultado obtenido: Correcto.	

Tabla 7.13: CP08. Crear una asignación.

Crear una asignación (Fecha anterior a la actual)	CP08-1
Descripción: Crear una asignación con una fecha de inicio anterior al día actual.	
Pasos: El adulto introduce una fecha inicial anterior al día de hoy.	
Resultado esperado: Se notifica al usuario que la fecha debe ser posterior a hoy.	
Resultado obtenido: Correcto.	

Tabla 7.14: CP08-1. Crear una asignación (Fecha anterior a la actual).

Crear una asignación (Formato de fecha incorrecto)	CP08-2
Descripción: Crear una asignación con un formato de fecha incorrecto.	
Pasos: El adulto introduce una fecha con un formato erróneo.	
Resultado esperado: Se notifica al usuario que el formato debe ser dd/mm/yyyy.	
Resultado obtenido: Correcto.	

Tabla 7.15: CP08-2. Crear una asignación (Formato de fecha incorrecto).

Modificar una asignación	CP09
Descripción: Modificar una asignación creada.	
Pasos: El adulto pulsa el botón de editar en la asignación correspondiente y modifica los datos.	
Resultado esperado: Se actualiza el documento en la base de datos y se actualizan las vistas.	
Resultado obtenido: Correcto.	

Tabla 7.16: CP09. Modificar una asignación.

Eliminar otros usuarios	CP10
Descripción: Eliminar otros miembros de la familia como adulto.	
Pasos: El adulto pulsa el botón de eliminar en el perfil correspondiente y confirma la eliminación en la ventana emergente.	
Resultado esperado: Se elimina el documento en la base de datos y todos sus datos asociados.	
Resultado obtenido: Correcto.	

Tabla 7.17: CP10. Eliminar otro usuario.

Eliminar una asignación	CP11
Descripción: Un adulto elimina una asignación.	
Pasos: El adulto pulsa el botón eliminar en la asignación correspondiente y confirma la eliminación en la ventana emergente.	
Resultado esperado: Se elimina el documento en la base de datos.	
Resultado obtenido: Correcto.	

Tabla 7.18: CP11. Eliminar una asignación.

Crear una tarea	CP12
Descripción: Un adulto crea una tarea.	
Pasos: El adulto pulsa el botón con un '+' en la sección de tareas, rellena los campos (mínimo descripción) y pulsa <i>Añadir</i> .	
Resultado esperado: Se crea el documento de esa tarea en la base de datos y se muestra en todas las listas de tareas (<i>widgets</i> incluidos).	
Resultado obtenido: Correcto.	

Tabla 7.19: CP12. Crear una tarea.

Eliminar tarea pendiente	CP13
Descripción: Un adulto elimina una tarea sin completar.	
Pasos: Un adulto pulsa en una tarea que todavía no se ha completado y confirma su eliminación en la ventana emergente.	
Resultado esperado: Se elimina el documento en la base de datos, se crea uno nuevo con esos datos en el historial y se elimina de todas las listas de tareas.	
Resultado obtenido: Correcto.	

Tabla 7.20: CP13. Eliminar tarea pendiente.

Añadir un gasto	CP14
Descripción: Un menor añade un nuevo gasto.	
Pasos: Un menor pulsa el botón <i>Añadir gasto</i> , introduce los datos y pulsa <i>Añadir</i> .	
Resultado esperado: Se crea un nuevo documento con los datos del gasto y se muestra en la lista de movimientos con los números en rojo.	
Resultado obtenido: Correcto.	

Tabla 7.21: CP14. Añadir un gasto.

Añadir un gasto (Dinero insuficiente)	CP14-1
Descripción: Añadir un gasto de valor mayor al dinero disponible.	
Pasos: Un menor añade un nuevo gasto con valor superior al dinero que tiene disponible.	
Resultado esperado: Se comunica al menor que no puede añadir gastos con un valor superior al dinero que tiene disponible.	
Resultado obtenido: Correcto.	

Tabla 7.22: CP14-1. Añadir un gasto (Dinero insuficiente).

Completar una tarea	CP15
Descripción: Marcar una tarea como completada.	
Pasos: Un menor pulsa el círculo de una una tarea para marcarla como completada.	
Resultado esperado: Se registra el menor que completó la tarea y se actualiza el estado.	
Resultado obtenido: Correcto.	

Tabla 7.23: CP15. Completar una tarea.

Reabrir una tarea	CP16
Descripción: Reabrir una tarea ya completada.	
Pasos: Un adulto pulsa una tarea completada y marca la opción de reabrir la tarea.	
Resultado esperado: Se actualiza el estado y se eliminan los datos de quién completó la tarea (si la tarea era para todos).	
Resultado obtenido: Correcto.	

Tabla 7.24: CP16. Reabrir una tarea.

Revisar y marcar tarea como completada	CP17
Descripción: Marcar una tarea como realmente completada.	
Pasos: Un adulto pulsa una tarea completada y marca la opción de cerrar la tarea.	
Resultado esperado: La tarea se elimina y se añade al historial. Si había una recompensa añadida se le da ese dinero al menor que la completó.	
Resultado obtenido: Correcto.	

Tabla 7.25: CP17. Revisar y marcar tarea como completada.

Crear un objetivo de ahorro	CP18
Descripción: Un menor crea un objetivo de ahorro.	
Pasos: Un menor pulsa el botón <i>Añadir objetivo</i> rellena todos los campos y pulsa <i>Añadir</i> .	
Resultado esperado: El objetivo se crea en la base de datos y se muestra el elemento visual correspondiente al objetivo.	
Resultado obtenido: Correcto.	

Tabla 7.26: CP18. Crear un objetivo de ahorro.

Ahorrar dinero	CP19
Descripción: Un menor ahorra para un objetivo.	
Pasos: Un menor pulsa el botón con el ‘cerdito’ y la flecha en verde, introduce la cantidad que desea ahorrar y pulsa el botón <i>Ahorrar</i> .	
Resultado esperado: Se resta ese dinero del dinero disponible del menor, se actualiza la cantidad ahorrada en base de datos y se muestra el progreso en ese objetivo.	
Resultado obtenido: Correcto.	

Tabla 7.27: CP19. Ahorrar dinero.

Retirar el dinero ahorrado	CP20
Descripción: Un menor retira el dinero ahorrado.	
Pasos: Un menor pulsa el botón con el ‘cerdito’ y la flecha en rojo, luego confirma que desea retirar el dinero.	
Resultado esperado: Se añade ese dinero al dinero disponible del menor, se crea un movimiento de tipo ingreso y se elimina el objetivo de base de datos.	
Resultado obtenido: Correcto.	

Tabla 7.28: CP20. Retirar el dinero ahorrado.

Jugar a un juego	CP21
Descripción: Un menor juega a uno de los juegos disponibles.	
Pasos: Un menor pulsa el botón del juego correspondiente y juega una partida.	
Resultado esperado: El juego se desarrolla sin ningún problema.	
Resultado obtenido: Correcto.	

Tabla 7.29: CP21. Jugar a un juego.

Recuperar la contraseña	CP22
Descripción: Un usuario ha olvidado la contraseña de la cuenta y quiere restablecerla.	
Pasos: El usuario pulsa en el texto de <i>Recuperar contraseña</i> e introduce el correo electrónico mediante el cual se registró.	
Resultado esperado: Se envía un correo con un enlace para restablecer la contraseña y se actualiza.	
Resultado obtenido: Correcto.	

Tabla 7.30: CP22. Recuperar la contraseña.

Modificar nombre de usuario desde mi perfil	CP23
Descripción: Un usuario quiero cambiar su nombre desde <i>mi perfil</i> .	
Pasos: El usuario accede al apartado <i>mi perfil</i> , pulsa en el botón de modificar nombre de usuario e introduce el nuevo nombre.	
Resultado esperado: Se actualiza el nombre en base de datos y en todos los elementos relacionados.	
Resultado obtenido: Correcto.	

Tabla 7.31: CP23. Modificar nombre de usuario desde mi perfil.

Modificar nombre de familia	CP24
Descripción: Un adulto quiere cambiar el nombre de la familia.	
Pasos: El adulto accede al apartado <i>mi perfil</i> , pulsa en el botón de modificar nombre de familia e introduce el nuevo nombre.	
Resultado esperado: Se actualiza el nombre en base de datos y en todos los elementos relacionados.	
Resultado obtenido: Correcto.	

Tabla 7.32: CP24. Modificar nombre de familia.

Modificar correo electrónico	CP25
Descripción: Un adulto quiere cambiar el correo asociado a la familia.	
Pasos: El adulto accede al apartado <i>mi perfil</i> , pulsa en el botón de modificar correo e introduce la nueva dirección.	
Resultado esperado: Se envía un correo de confirmación a la nueva dirección.	
Resultado obtenido: Correcto.	

Tabla 7.33: CP25. Modificar correo electrónico.

Modificar contraseña del perfil	CP26
Descripción: Un usuario quiere establecer una contraseña para su perfil.	
Pasos: El usuario accede al apartado <i>mi perfil</i> , pulsa en el botón de modificar contraseña e introduce la nueva contraseña. Si ya existía una contraseña es necesario introducirla de nuevo.	
Resultado esperado: Se establece dicha contraseña para ese perfil y se almacena encriptada en la base de datos.	
Resultado obtenido: Correcto.	

Tabla 7.34: CP26. Modificar contraseña del perfil.

Establecer foto de perfil	CP27
Descripción: Un usuario quiere tener una imagen para su perfil.	
Pasos: El usuario accede al apartado <i>mi perfil</i> , pulsa en el botón con un icono de cámara y hace una fotografía con la cámara integrada o la añade desde su galería.	
Resultado esperado: Se añade esa imagen al <i>Storage</i> de <i>Firebase</i> y se actualiza la imagen de perfil en todas las pantallas.	
Resultado obtenido: Correcto.	

Tabla 7.35: CP27. Establecer foto de perfil.

Eliminar datos de la familia	CP28
Descripción: Un adulto desea eliminar la cuenta de la familia.	
Pasos: El adulto pulsa el botón <i>Eliminar cuenta</i> en la ventana de <i>mi perfil</i> y confirma introduciendo la contraseña de la cuenta.	
Resultado esperado: Si no existen otros perfiles se elimina la cuenta y todos los datos asociado a esta. Si existen otros perfiles de adulto se elimina este perfil. Si quedan perfiles de menores, se informa de que se deben eliminar sus perfiles previamente.	
Resultado obtenido: Correcto.	

Tabla 7.36: CP28. Eliminar datos de la familia.

Repetir tarea desde el historial	CP29
Descripción: Un adulto desea repetir una tarea desde el historial.	
Pasos: El adulto accede al historial y pulsa el botón con el icono de repetir, puede asignarla a un menor desde la ventana emergente y confirmar.	
Resultado esperado: La tarea se borra del historial, se añade a la lista de tareas y se borra la fecha de completado.	
Resultado obtenido: Correcto.	

Tabla 7.37: CP29. Repetir tarea desde el historial.

Consultar movimientos de un menor	CP30
Descripción: Un adulto desea consultar los movimientos de uno de los menores de la familia.	
Pasos: El adulto selecciona el nombre del menor en la pantalla de <i>Consultar movimientos</i> .	
Resultado esperado: Si el menor tiene movimientos se muestran justo debajo del selector y en caso contrario muestra el mensaje: No existen movimientos.	
Resultado obtenido: Correcto.	

Tabla 7.38: CP30. Consultar movimientos de un menor.

Capítulo 8

Conclusiones

8.1. Conclusiones

Como resultado de este trabajo de fin de grado, se ha desarrollado una aplicación móvil que permite a las familias introducir a los niños en los conceptos básicos de la gestión del dinero. Al repasar los objetivos e hitos del proyecto, podemos ver que estos han sido alcanzados, consiguiendo los siguientes logros para cada uno de ellos:

- *Planificación del proyecto y establecimiento de objetivos.*
 - Se planificó detalladamente todo el proyecto, definiendo las metodologías de trabajo y evaluando los riesgos asociados.
 - Se realizó un análisis de estudios y publicaciones sobre educación para orientar correctamente la aplicación.
 - Se estudió el mercado para identificar posibles competidores y analizar sus propuestas para abordar nuestro problema.
 - Se elaboraron los requisitos y los diagramas UML de la aplicación basándose en la información recopilada.
- *Diseño de la aplicación.*
 - Se estudió cómo debería ser una interfaz adecuada para la aplicación y se creó un prototipo.
 - Se diseñó una arquitectura software adaptada a nuestras necesidades.
- **Objetivo principal:** *desarrollar una aplicación móvil educativa diseñada para enseñar a los niños las habilidades fundamentales de administración financiera, incluyendo la gestión del dinero y el ahorro, todo ello de una manera dinámica y amena.*
 - La aplicación permite a los adultos asignar propinas y supervisar los gastos de los menores, permitiendo un seguimiento de su aprendizaje.

- Los menores pueden anotar gastos y establecer objetivos de ahorro, fomentando el aprendizaje de buenos hábitos financieros.
- La aplicación incluye un juego sencillo para que los menores mejoren sus habilidades matemáticas y aprendan conceptos económicos.
- La aplicación incorpora un sistema de tareas para organizar el trabajo dentro del hogar.

En lo que respecta a la adquisición de nuevos conocimientos por parte del alumno:

- Se han aprendido y aplicado nuevas tecnologías para el desarrollo de aplicaciones móviles.
- Se han mejorado y puesto en práctica habilidades del diseño de software.
- Se ha adquirido experiencia en la planificación de proyectos y se ha hecho frente a los problemas que surgen durante un desarrollo.

Como conclusión final, la principal aportación de este proyecto es una aplicación destinada a complementar la educación financiera de los niños dentro del entorno familiar. Aunque aún tiene margen de mejora, su empleo dentro del hogar fomentará el desarrollo de buenos hábitos financieros entre los más pequeños. Las limitaciones del trabajo han sido numerosas, siendo la más significativa la imposibilidad de realizar pruebas con un número mayor de familias. Sin embargo, gracias a Juan, investigador en tecnología educativa, y gracias a las investigaciones realizadas previamente al desarrollo, la aplicación cuenta con una orientación adecuada.

8.2. Planes de futuro

Como planes de futuro, se dejan planteadas las siguientes propuestas, que debido al tiempo limitado asignado a un TFG no ha sido posible desarrollarlas:

- **Empleo de dinero real:** unas de las ideas iniciales para la aplicación era que esta no fuera únicamente una simulación o una libreta donde llevar las cuentas, si no que funcionase también como una tarjeta de prepago o *monedero*. Esta funcionalidad implica una serie de aspectos extra como: considerar las diferentes regulaciones, mejorar la seguridad de la aplicación y llegar a acuerdos con instituciones financieras, entre otros, por lo que se salía del alcance del proyecto establecido.
- **Refactorización del código:** para este proyecto, se optó por seguir la arquitectura MVC (Modelo-Vista-Controlador) debido al tiempo disponible, a la familiaridad del alumno con esta y a que era la empleada en la formación previa. Sin embargo, la arquitectura MVVM (Model-View-ViewModel) ha ganado popularidad por su mayor modularidad y facilidad de mantenimiento del código, al separar las capas de presentación y lógica entre View y ViewModel, respectivamente.

Aunque se reconoce el beneficio de migrar a MVVM, una refactorización completa llevaría de dos a cuatro semanas. Por lo tanto, se deja planteada como una mejora futura para la aplicación.

- **Envío de notificaciones *push* a todos los dispositivos de la familia:** en la versión actual de la aplicación, cuando un adulto activa las notificaciones *push*, recibe una notificación cada vez que un menor añade un gasto. Sin embargo, estas notificaciones solo se envían al dispositivo que está siendo utilizado en ese momento. Se había planeado incluir la funcionalidad de enviar notificaciones a todos los dispositivos donde la familia tenga una sesión iniciada, pero debido a limitaciones de tiempo, esta característica se quedó fuera del alcance del proyecto.
- **Pruebas con más usuarios:** una de las limitaciones de este trabajo ha sido no contar con más niños de las edades para las que se ha diseñado la aplicación. Para probar mejor los resultados con usuarios reales, sería ideal llegar a algún acuerdo con un colegio para realizar una actividad didáctica entre padres y niños, observando sus opiniones, quejas, reacciones, etc.
- **Salida a producción:** se planea publicar la aplicación en la Play Store de forma gratuita y por un tiempo limitado para evaluar su acogida entre el público. Si la aplicación tuviera mucho éxito, sería necesario considerar aspectos como la monetización o la migración del servicio *back-end* en caso de que las cuotas que ofrece Firebase resulten demasiado elevadas.

8.3. Valoración personal

Personalmente estoy muy orgulloso del trabajo que he realizado junto a la ayuda de mi tutor Juan. Inicialmente este TFG comenzó con una idea inicial y sin rumbo fijo, y el trabajo que tenía por delante parecía inabarcable, pero el resultado final supera mis expectativas iniciales. Este proyecto me ha enseñado que, aunque un proyecto pueda parecer imposible al principio, con una organización adecuada se puede lograr. Además, he aprendido un nuevo lenguaje de programación y he mejorado mis habilidades y conocimientos en el desarrollo de software. Sin embargo, eso no se compara con la satisfacción de poder aplicar lo que he aprendido a lo largo de la carrera y llevar a cabo un proyecto que, además, es propio.

Bibliografía

- [1] Instituto Nacional de Evaluación Educativa. *PISA 2018 Competencia financiera*. Inf. téc. 2018.
- [2] Instituto Santalucía. *Documental “Y a mí qué el dinero”*. <https://www.youtube.com/watch?v=szXfLtmlDPo>. Oct. de 2021.
- [3] Elsa Fornero y Anna Lo Prete. “Financial education: From better personal finance to improved citizenship”. En: *J. financ. lit. wellbeing* 1.1 (2023), págs. 12-27.
- [4] Alessandro Buccioli y Marcella Veronesi. “Teaching children to save: What is the best strategy for lifetime savings?” En: *J. Econ. Psychol.* 45 (2014), págs. 1-17.
- [5] Martin; Oliva Allen. “Teaching children about money: Applications of social learning and cognitive learning development theories”. En: 93.2 (2001), págs. 26-29.
- [6] *El uso de las tecnologías por los menores en España*. <https://www.ontsi.es/es/publicaciones/uso-nuevas-tecnologias-menores-Espana-2022>. Accessed: 2023-12-6.
- [7] idDOCENTE. *Uso del móvil en el aula: a favor o en contra?* <https://iddocente.com/uso-del-movil-aula-a-favor-en-contra/>. Accessed: 2023-12-31. Ene. de 2022.
- [8] Javier Fombona y Pablo Roza Martin. “Uso de los dispositivos móviles en educación infantil”. En: *EDMETIC* 5.2 (2016), pág. 158.
- [9] *Global mobile consumer trends*. <https://www2.deloitte.com/do/es/pages/technology-media-and-telecommunications/articles/pr-global-mobile-consumer-trends.html>. Accessed: 2024-2-27. Mayo de 2016.
- [10] *Duolingo: Aprende inglés, francés y otros idiomas gratis*. <https://es.duolingo.com/>. Accessed: 2023-12-31.
- [11] *Udemy*. <https://www.udemy.com/>. Accessed: 2023-12-31.
- [12] Diana Moreno. *Se dispara el uso de pantallas en niños y niñas durante el confinamiento*. <https://ciudadesamigas.org/pantallas-infancia-cuarentena/>. Accessed: 2024-1-21. Jun. de 2020.
- [13] *¿Premiar con pantallas? Tecnología: premios que generan dependencia*. <https://www.magisnet.com/2022/01/premiar-con-pantallas-tecnologia-premios-que-generan-dependencia/>. Accessed: 2024-2-9. Oct. de 2019.

- [14] *Our money habits are largely set by the age of 7*. <https://www.redstareducation.co.uk/blog/our-money-habits-are-largely-set-by-the-age-of-7/>. Accessed: 2023-12-8. Ago. de 2020.
- [15] Mark C Schug. "Economics for Kids: Ideas for Teaching in the Elementary Grades". En: *Meeting the standards: Social studies readings for K-6 educators*. ERIC, 1986, págs. 165-167.
- [16] Rocío Escobar. *Etapas del desarrollo cognitivo - Teoría de Piaget*. <https://www.forobebe.com/etapas-desarrollo-cognitivo-piaget/>. Accessed: 2024-1-21. Abr. de 2016.
- [17] *Teaching kids the value of money*. <https://www.cambridge-credit.org/teaching-kids-the-value-of-money.html>. Accessed: 2023-12-8.
- [18] Argenis Montilla Wilmarys González. "Revista Talento. Vol. 1: Estrategia de enseñanza de conceptos básicos de educación financiera para niños en edad escolar: Strategy for teaching basic concepts of financial education for children of school age". En: (2019), págs. 26-39.
- [19] Melanie D Jewkes. *Teaching children money management*. https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=1149&context=extension_curall. Accessed: 2023-12-31.
- [20] *Cómo enseñar educación financiera a niños y niñas*. <https://www.educo.org/blog/como-ensenar-educacion-financiera-a-ninos>. Accessed: 2023-12-7. Ene. de 2022.
- [21] Sean Brotherson Debra Pankow. *Talking to Children About Money*. https://library.ndsu.edu/ir/bitstream/handle/10365/9331/fs1441_2009.pdf?sequence=1. [Accessed 21-01-2024].
- [22] *¿Sabes qué es la educación financiera? Aplica estas 4 claves — mariatrusa.org*. <https://mariatrusa.org/blog/sabes-que-es-la-educacion-financiera-aplica-estas-4-claves/>. [Accessed: 2024-01-05].
- [23] Redes Sociales. *Paga de los niños: ómo darla, cómo empezar y con qué frecuencia?* <https://www.bankinter.com/blog/finanzas-personales/dar-paga-hijos>. Accessed: 2023-12-6. Sep. de 2020.
- [24] *Prepaid debit cards for kids, teens, families*. <https://famzoo.com/>. Accessed: 2023-12-8.
- [25] *Mi dinero y Yo*. https://play.google.com/store/apps/details?id=app.midineroyyo.com&hl=es_MX&pli=1. Accessed: 2023-12-8.
- [26] *Money Kids - Count real money*. <https://play.google.com/store/apps/details?id=io.ionic.moneykids>. Accessed: 2023-12-8.
- [27] *The game of life 2*. <https://play.google.com/store/apps/details?id=com.marmalade.gol2&hl=es&gl=US&pli=1>. Accessed: 2024-1-24.
- [28] *MONOPOLY*. https://play.google.com/store/apps/details?id=com.marmalade.monopoly&hl=es_419&gl=US. Accessed: 2024-1-24.
- [29] <https://bankaroo.com/>. Accessed: 2024-1-24.
- [30] *NatWest Rooster Money*. <https://roostermoney.com/>. Accessed: 2024-1-24. Jul. de 2023.

- [31] David J Anderson y Andy Carmichael. *Essential Kanban Condensed*. Lean-Kanban University, 2016.
- [32] Valeria S Meaurio y Eric Schmieder. “La Arquitectura de Software en el Proceso de Desarrollo: Integrando MDA al Ciclo de Vida en Espiral”. En: *Rev. Latinoam. Ing. Softw.* 1.4 (2014), págs. 142-146.
- [33] Administración y Evaluación de Proyectos. *Modelo Espiral de un proyecto de desarrollo de software*. <https://www.ojovisual.net/galofarino/modeloespiral.pdf>. Accessed: 2023-12-8.
- [34] Project Management Institute. *Guía de los Fundamentos Para la Dirección de Proyectos (Guía del PMBOK): (Spanish version of: A Guide to the Project Management Body of Knowledge: PMBOK Guide)*. 6.ª ed. Newton Square, PA, Estados Unidos de América: Project Management Institute, 2017.
- [35] *Sueldo promedio desarrollador Android*. <https://es.indeed.com/career/android-developer/salaries>. Accessed: 2023-12-30.
- [36] *Overleaf - editor online LaTeX*. <https://www.uc3m.es/sdic/services/overleaf>. Accessed: 2023-12-31.
- [37] *Figma: The Collaborative Interface Design Tool*. <https://www.figma.com/>. Accessed: 2023-12-31.
- [38] *¿Qué es figma?* <https://cei.es/que-es-figma/>. Accessed: 2023-12-31. Sep. de 2021.
- [39] *Flowchart maker & online diagram software*. <http://draw.io/>. Accessed: 2023-12-31.
- [40] *Draw.io - Google workspace marketplace*. <https://workspace.google.com/marketplace/app/drawio/671128082532?hl=es>. Accessed: 2023-12-31.
- [41] *Gestiona los proyectos de tu equipo desde cualquier lugar*. <https://trello.com/es>. Accessed: 2023-12-31.
- [42] *Qué es Trello: descubre sus funciones, usos y todo lo que ofrece*. <https://trello.com/es/tour>. Accessed: 2023-12-31.
- [43] Atlassian. *Qué es Git*. <https://www.atlassian.com/es/git/tutorials/what-is-git>. Accessed: 2023-12-31.
- [44] *GitHub*. <https://github.com/>. Accessed: 2023-12-31.
- [45] Antonio Muñoz. *¿ es GitHub y para qué sirve?* <https://www.webempresa.com/hosting/que-es-github.html>. Accessed: 2023-12-31. Sep. de 2020.
- [46] Atlassian. *Sourcetree*. <https://www.sourcetreeapp.com/>. Accessed: 2024-6-6.
- [47] *Descarga e instala Android Studio*. <https://developer.android.com/codelabs/basic-android-kotlin-compose-install-android-studio?hl=es-419>. Accessed: 2024-2-20.
- [48] *Curso Android Studio Online*. <https://www.formadoresit.online/cursos/desarrollo-y-programacion/aplicaciones-moviles/curso-android-studio-online/>. Accessed: 2024-2-20.
- [49] Mati Presta. *¿Qué es el backend de Firebase?* <https://blog.back4app.com/es/que-es-el-backend-de-firebase>. [Accessed: 2024-04-31].
- [50] *Interfaz de usuario (UI) - Bisiesto Estudio — bisiesto.es*. <https://bisiesto.es/interfaz-de-usuario-ui/>. [Accessed: 2024-06-07].

- [51] *Children's UX: Usability issues in designing for young people*. <https://www.nngroup.com/articles/childrens-websites-usability-issues/>. Accessed: 2023-12-30.
- [52] Fabián Cordero y Universidad del Azuay - Ecuador. "DISEÑO DE INTERFACES GRÁFICAS PARA RECURSOS DIDÁCTICOS DIGITALES". En: *DAYA. Diseño, Arte y Arquitectura*. 1.5 (2018), págs. 11-29.
- [53] Jakob Nielsen. *10 usability heuristics for user interface design*. <https://www.nngroup.com/articles/ten-usability-heuristics/>. Accessed: 2024-2-27. Abr. de 1994.
- [54] G Sánchez Morales, C Mezura-Godoy y E Benítez-Guerrero. "Proceso de pruebas de usabilidad de software: Software usability testing process". En: *Tecnología Educativa Revista CONAIC* 5.1 (2018), págs. 27-32.
- [55] Javier Arias Del Prado. *Test con usuarios de guerrilla - UXABLES — Blog — uxables.com*. <https://www.uxables.com/investigacion-ux/test-con-usuarios-de-guerrilla/>. [Accessed 07-06-2024].
- [56] Juan Ramos. *Android: ¿Qué es MVC, MVP y MVVM? — programacionymas.com*. <https://programacionymas.com/blog/android-mvc-mvp-mvvm/>. [Accessed 09-06-2024].
- [57] Inc. Cloudflare. *¿Qué es BaaS? — Backend como servicio vs. sin servidor*. <https://www.cloudflare.com/es-es/learning/serverless/glossary/backend-as-a-service-baas/>. [Accessed: 2024-05-31].
- [58] Microsoft. *¿Qué es el desarrollo de aplicaciones móviles?* <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-mobile-app-development/>. [Accessed: 2024-05-31].
- [59] Álvaro Javier Duran Sanjuan y Jorge Luis Peinado Rodríguez. "Confrontación de dos tecnologías de desarrollo de aplicaciones móviles sobre la plataforma Android 4.0 desde la perspectiva de los atributos de calidad." En: (2014), págs. 124-131.
- [60] *SQL vs NoSQL: Ventajas y Desventajas Detalladas de Cada Sistema — dongee.com*. <https://www.dongee.com/tutoriales/sql-vs-nosql-ventajas-y-desventajas-detalladas-de-cada-sistema/>. [Accessed 09-06-2024].
- [61] *Adapter — refactoring.guru*. <https://refactoring.guru/es/design-patterns/adapter/>. [Accessed 09-06-2024].
- [62] andijakl. *Kotlin & RecyclerView for High Performance Lists in Android — andreasjakl.com — andreasjakl.com*. <https://www.andreasjakl.com/kotlin-recyclerview-for-high-performance-lists-in-android/>. [Accessed 09-06-2024].
- [63] *Singleton — refactoring.guru*. <https://refactoring.guru/es/design-patterns/singleton/>. [Accessed 09-06-2024].
- [64] Atlassian. *Gitflow Workflow — Tutorial de Gitflow — Atlassian Git Tutorial*. Accessed: 2024-04-11. n.d. URL: <https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow>.

Apéndice A

Manual de instalación

Para instalar la aplicación, se requiere un dispositivo con el sistema operativo Android, cuya versión no debe ser inferior a Android 9. El enlace esta disponible en el apéndice D. Para instalar la aplicación solo se deben seguir los pasos descritos en la figura A.2.

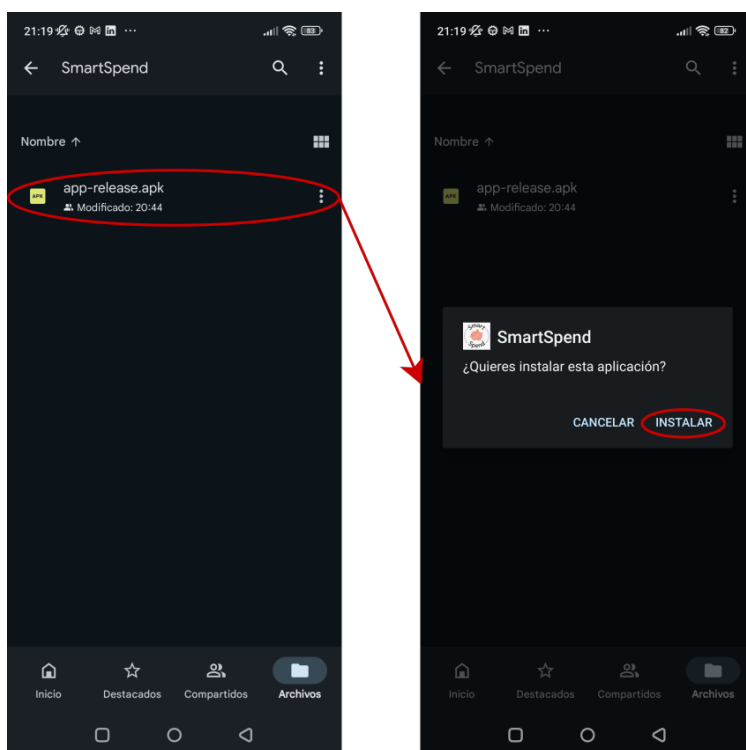


Figura A.1: Pasos para instalar la aplicación

La mayoría de los dispositivos Android solo permiten la instalación de aplicaciones desde la Play Store por defecto. Debes modificar estos ajustes desde la opción *Instalar aplicaciones desconocidas* en la configuración de tu dispositivo.

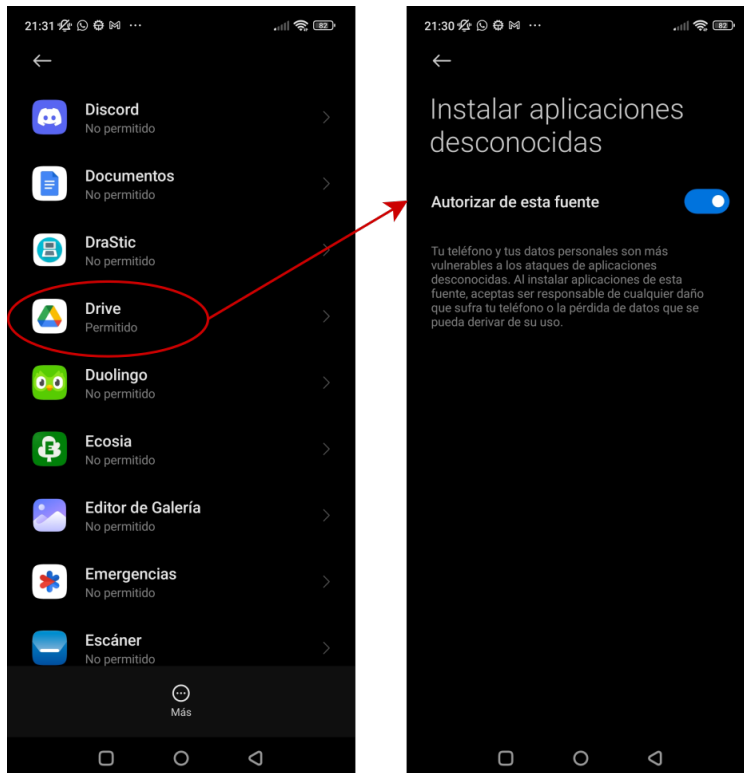


Figura A.2: Autorizar descargas desde Drive

Apéndice B

Manual de usuario

En este apéndice se describe el funcionamiento de la aplicación para los usuarios. Se mostrarán diferentes capturas de la aplicación a través de las cuales se describirán los principales flujos de acciones y funcionalidades.

B.1. Inicio de sesión y registro en la aplicación

Los usuarios pueden acceder a la aplicación mediante correo electrónico y contraseña. Para ello previamente se deben registrar y confirmar su email mediante el correo que se envía al completar el registro. Si desean probar la aplicación sin necesidad de registrarse pueden acceder mediante el modo invitado (pulsando en *Acceder como invitado*), sin embargo, los datos en este modo no se mantendrán entre sesiones. Finalmente, si se olvidan de su contraseña solo tiene que pulsar en *Recuperar contraseña* e introducir su email para que les llegue un correo con el enlace para restablecerla (Figura B.1).

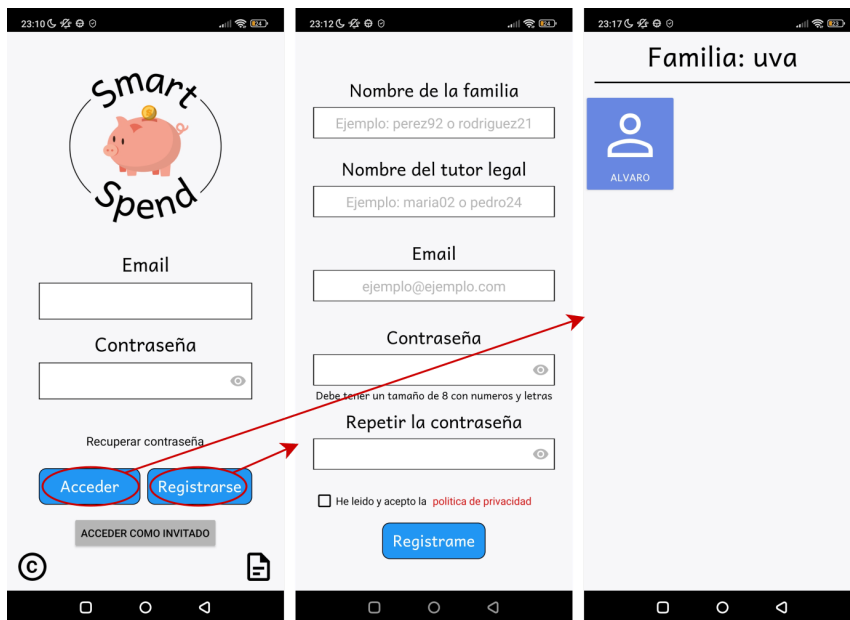


Figura B.1: Pantalla de inicio, registro y perfiles

B.2. Establecer contraseña de perfil

En la aplicación existen dos tipos de contraseñas: la de la cuenta y las de los perfiles. La contraseña de la cuenta permite a una familia iniciar sesión en un dispositivo, mientras que las contraseñas de perfil sirven para bloquear el acceso a perfiles específicos. Esto es importante porque los perfiles de adulto pueden realizar operaciones críticas como borrar usuarios o asignar dinero a los menores. De esta manera, se impide que un menor acceda a otros perfiles que no sean el suyo.

Al crear una cuenta, el perfil de adulto comienza sin contraseña. Debido a que las operaciones permitidas a un adulto son críticas, se recomienda establecer una contraseña desde el perfil. Para acceder al perfil, haga clic en su nombre e imagen en la esquina superior derecha o desde el menú emergente en la esquina opuesta. Una vez dentro del perfil, seleccione *Editar contraseña* (Figura B.2). Si en algún momento el usuario olvida la contraseña de su perfil, puede introducir la contraseña de la cuenta para eliminar la contraseña y acceder a su pantalla principal (Figura B.3).

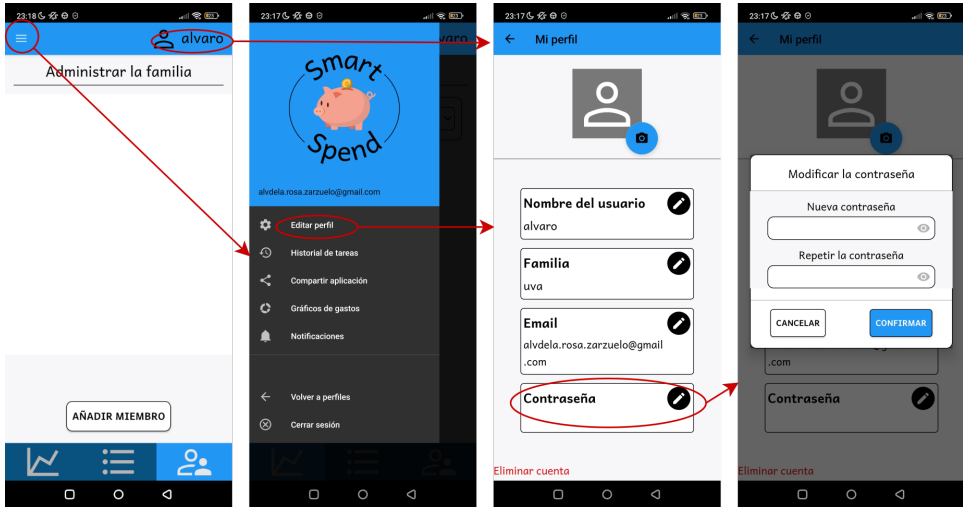


Figura B.2: Pasos para modificar la contraseña del perfil



Figura B.3: Acceder al perfil si se ha olvidado la contraseña

B.3. Crear nuevos miembros de la familia

Como perfil adulto puede ir a la sección de administración (tercer botón inferior) y añadir nuevos miembros a la cuenta. Debe seleccionar un tipo de perfil entre adulto y menor y establecer un nombre para ese usuario. Posteriormente, la contraseña es opcional y si es un perfil de menor se puede establecer un dinero inicial para que comience a gestionarlo (Figura B.4).

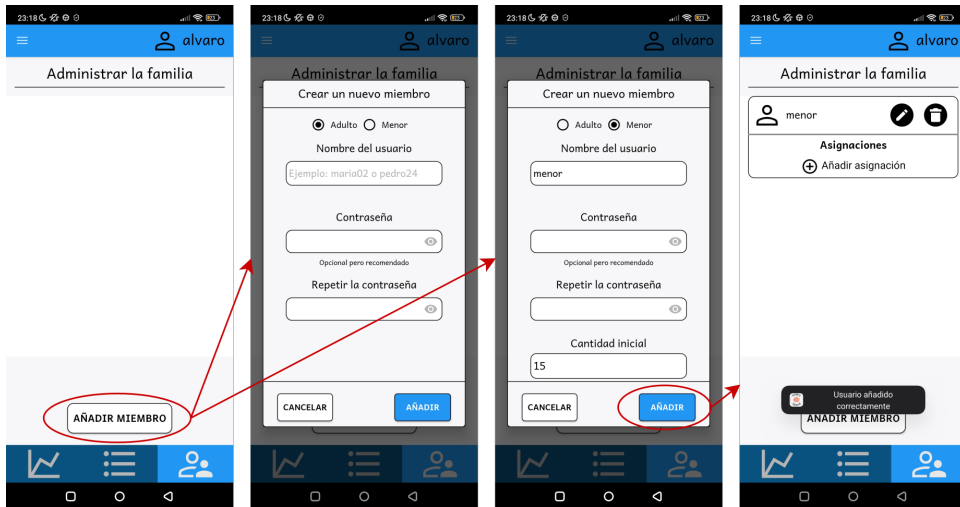


Figura B.4: Creación del perfil de un menor

B.4. Crear una asignación

Para crear la asignación de un menor, pulse el botón *Añadir asignación* debajo del perfil del menor en la sección de administración. Introduzca un nombre, una fecha de inicio, el tipo (que define la frecuencia) y la cantidad. Siempre puede editar o eliminar una asignación usando los botones correspondientes en el elemento creado (Figura B.5).

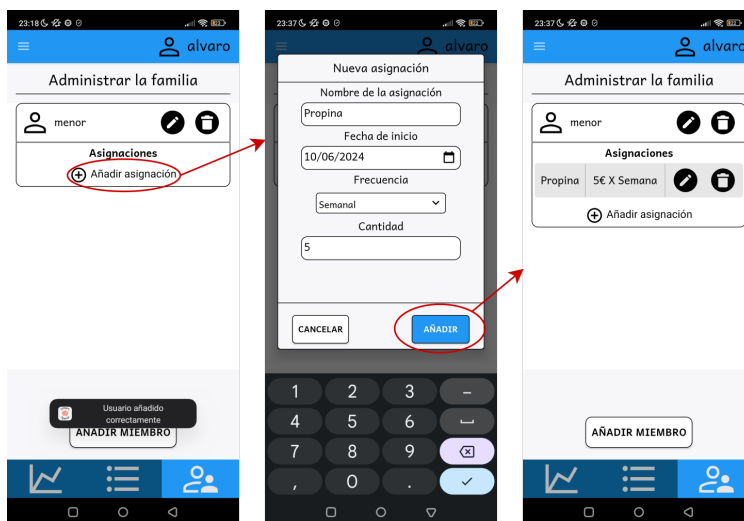


Figura B.5: Creación de una asignación

B.5. Gestión de las tareas

Un adulto puede crear tareas desde la sección de tareas pulsando el botón con un '+'. La descripción es obligatoria. Después, se puede establecer una fecha límite, determinar si la tarea es obligatoria u opcional, añadir una recompensa y/o asignársela a un menor de la familia (Figura B.6).

Las tareas creadas se mostrarán en la sección de tareas del menor, donde puede marcar el círculo para señalarlas como completadas. Posteriormente, el adulto debe verificar si la tarea se ha realizado correctamente o si necesita volver a abrirla. Una vez completadas, las tareas pasan al historial, desde donde se pueden repetir o borrar todo el historial (Figura B.7).

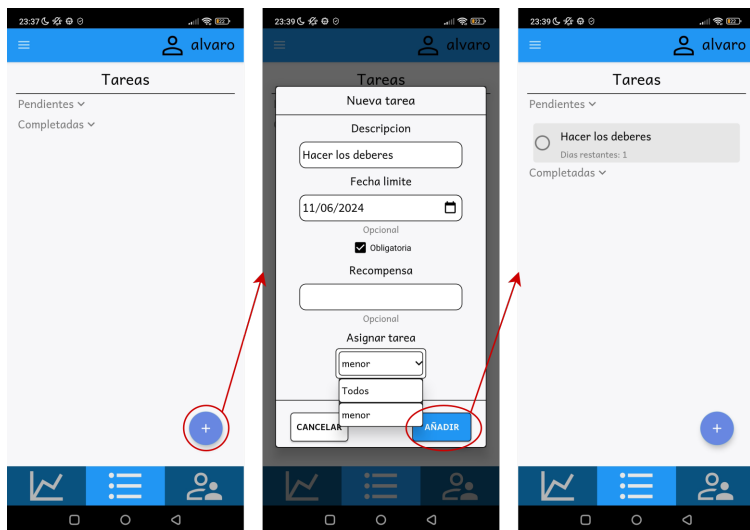


Figura B.6: Creación de una nueva tarea

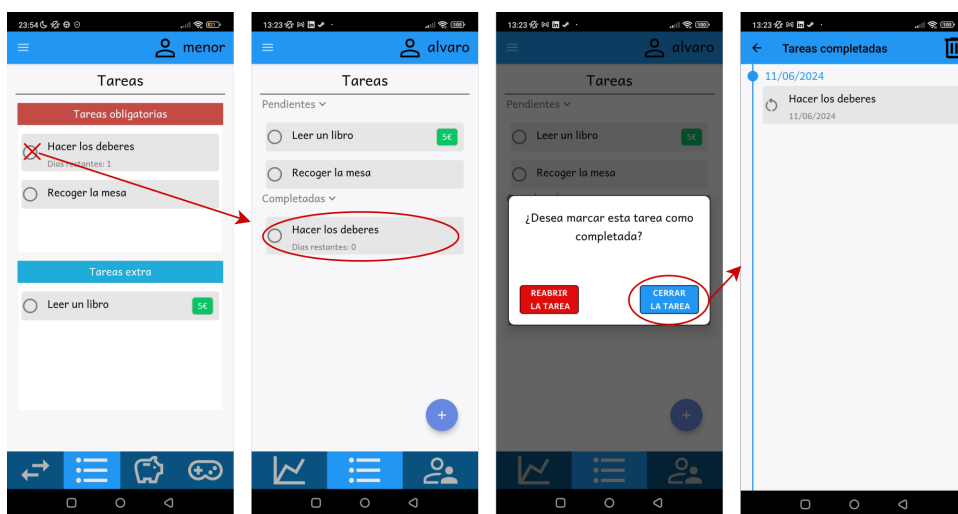


Figura B.7: Flujo de control de las tareas

B.6. Añadir gastos

Desde su perfil, los menores pueden añadir y controlar sus gastos diarios. Para ello, deben pulsar el botón *Añadir gasto* y escribir los campos de descripción y cantidad (el tipo es opcional pero recomendable). La lista de movimientos se mostrará justo debajo del botón (Figura B.8).

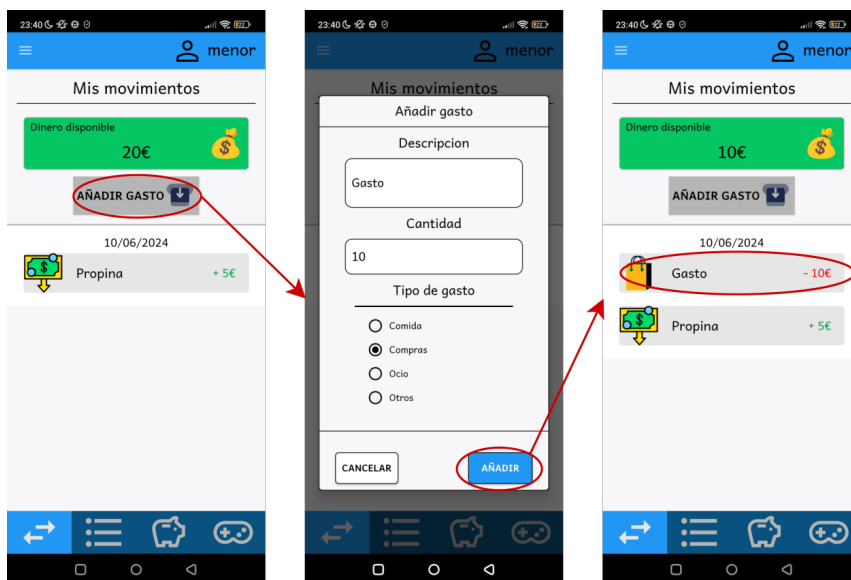


Figura B.8: Flujo para añadir un gasto

B.7. Consultar movimientos de dinero de los menores

Los adultos pueden consultar en todo momento los gastos e ingresos de los menores desde su ventana. En la primera pantalla que se muestra al acceder, solo deben escoger al menor cuyos movimientos desean ver. Además, desde el menú emergente, pueden acceder a la sección de gráficos para visualizar los datos en forma de gráficos (tipos de gastos y gastos mensuales) (Figura B.9).

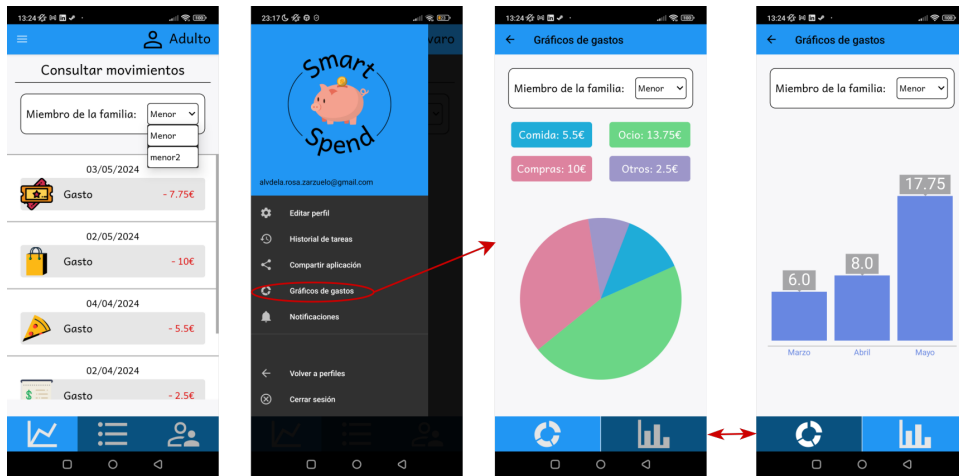


Figura B.9: Formas de consultar los movimientos de dinero de un menor

B.8. Crear y alcanzar objetivos de ahorro

Los menores pueden crear objetivos de ahorro pulsando *Añadir objetivo*. Deben describir para qué quieren ahorrar y el precio del objetivo. Una vez creado, pueden ahorrar dinero pulsando el botón con una flecha verde o retirar dinero pulsando el botón con una flecha roja (Figura B.10). El objetivo se actualiza según el dinero ahorrado, mostrando visualmente el progreso (Figura B.11). El dinero se puede retirar en cualquier momento, pero al alcanzarse el objetivo, este se pinta de verde y ya no se puede pulsar el botón de ahorrar (Figura B.12).

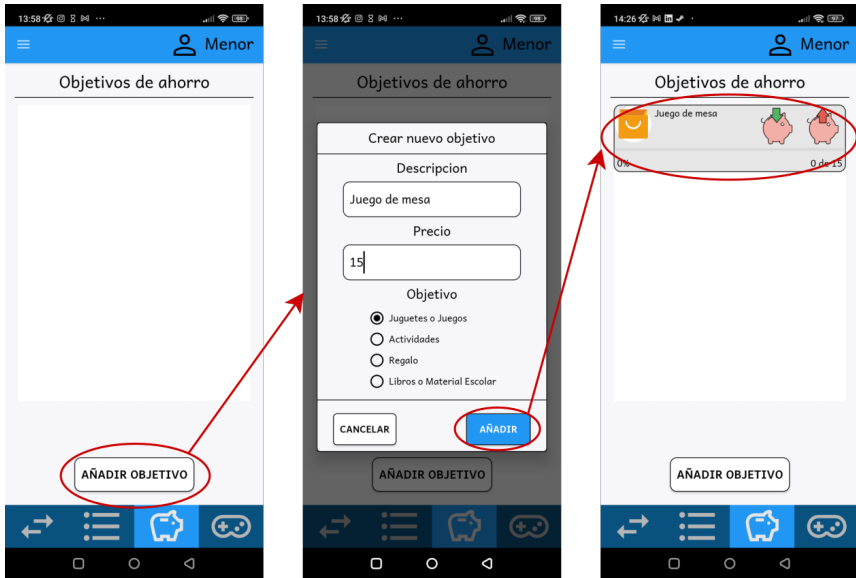


Figura B.10: Crear un objetivo de ahorro

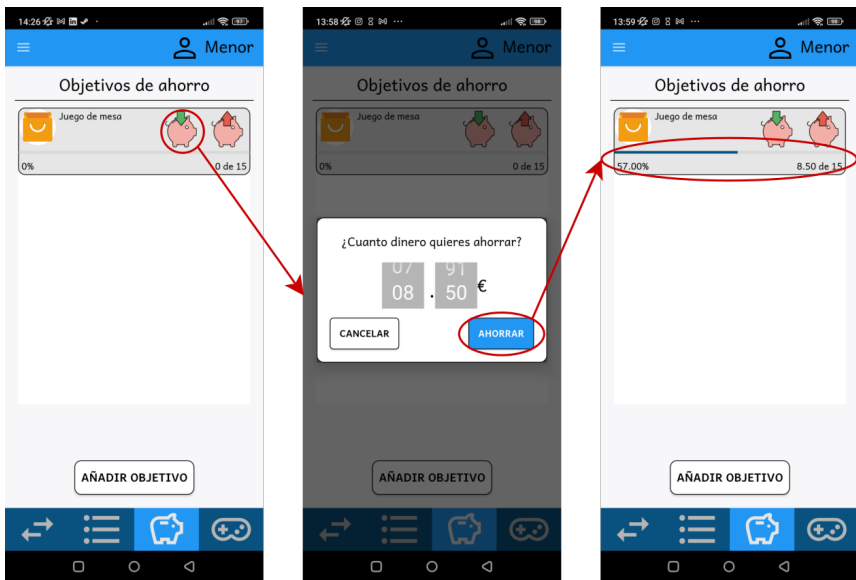


Figura B.11: Ahorrar dinero para un objetivo

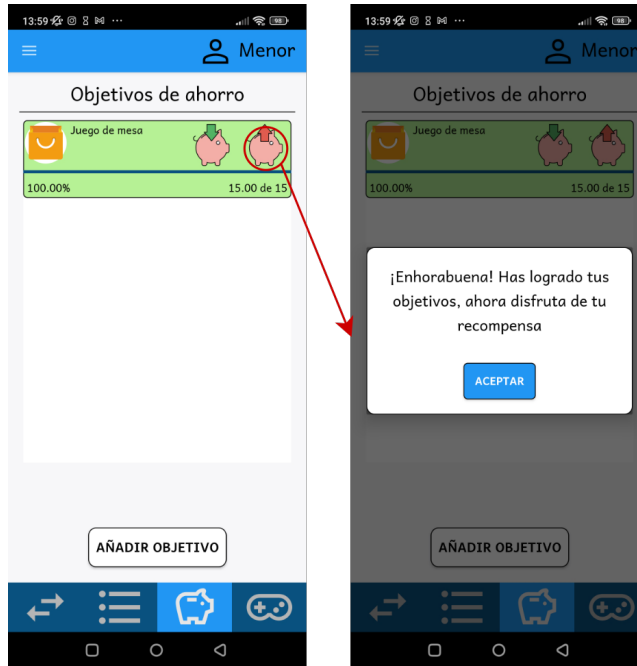


Figura B.12: Objetivo alcanzado y retirar el dinero ahorrado

Apéndice C

Política de privacidad

Este apéndice incluye la política de privacidad de la aplicación.

Esta Política de Privacidad describe nuestras políticas y procedimientos sobre la recopilación, uso y divulgación de su información cuando utiliza la aplicación y le informa sobre sus derechos de privacidad y cómo la ley lo protege.

Utilizamos sus datos personales para proporcionar y mejorar el servicio. Al utilizar el servicio, acepta la recopilación y el uso de información de acuerdo con esta Política de Privacidad. Esta Política de Privacidad ha sido creada con la ayuda del Generador de Políticas de Privacidad. (<https://www.privacypolicies.com/es/>)

Interpretación y Definiciones

Interpretación

Las palabras cuya inicial está en mayúscula tienen significados definidos bajo las siguientes condiciones. Las siguientes definiciones tendrán el mismo significado independientemente de si aparecen en singular o en plural.

Definiciones

A efectos de esta política de privacidad:

- **Cuenta:** significa una cuenta única creada para que usted acceda a nuestro Servicio.
- **Aplicación:** se refiere a SmartSpend, el programa de software proporcionado por el estudiante.

- **País:** se refiere a España.
- **Dispositivo:** significa cualquier dispositivo que pueda acceder a la aplicación, como un teléfono móvil o una tableta digital.
- **Datos personales:** es cualquier información que se refiere a un individuo identificado o identificable.
- **Datos de uso:** se refiere a datos recopilados automáticamente, generados por el uso de la aplicación.
- **Usted:** significa el individuo que accede o utiliza la aplicación.

Recopilación y uso de datos personales

Tipos de datos recopilados

Datos personales

Mientras utiliza nuestra aplicación es posible que le pidamos que nos proporcione cierta información que se pueda utilizar para contactarlo o identificarlo. La información personalmente identificable puede incluir, entre otros:

- Correo electrónico

Datos de uso

Los datos de uso pueden incluir información como el modo de uso de la aplicación, la hora y la fecha de uso, el tiempo que pasa en la aplicación, y otros datos de diagnóstico.

Cuando accede a la aplicación a través de un dispositivo móvil, es posible que recopilemos cierta información automáticamente, incluido, entre otros, el tipo de dispositivo móvil que utiliza o su sistema operativo móvil, únicamente con el objetivo de mejorar la experiencia del usuario.

Información Recopilada Mientras Usa la Aplicación

Mientras usa nuestra aplicación, para proporcionar funciones de nuestra Aplicación, podemos recopilar, con su permiso previo:

- Fotografías y otra información de la cámara y la biblioteca de fotos de su dispositivo.

Utilizamos esta información para proporcionar funciones de nuestra aplicación. Puede habilitar o deshabilitar el acceso a esta información en cualquier momento, a través de la configuración de su dispositivo.

Uso de los datos personales

Usaremos los Datos Personales para los siguientes propósitos: para proporcionar y mantener nuestra aplicación, para administrar su cuenta, para contactarlo, para administrar sus solicitudes.

Podemos compartir su información personal en las siguientes situaciones:

- Con Proveedores de Servicios: podemos compartir su información con nuestros Proveedores de Servicios, en este caso Firebase.
- Con su consentimiento: podemos divulgar su información personal para propósitos académicos, únicamente con su consentimiento.

Retención de datos personales

Retendremos sus Datos Personales solo durante el tiempo que sea necesario para los fines establecidos en esta Política de Privacidad. Retendremos y utilizaremos sus Datos Personales en la medida necesaria para cumplir con nuestras obligaciones legales (por ejemplo, si estamos obligados a retener sus datos para cumplir con las leyes aplicables), resolver disputas y hacer cumplir nuestros acuerdos y políticas legales.

También retendrá Datos de Uso con fines de análisis. Los Datos de Uso generalmente se retienen por un período de tiempo más corto, excepto cuando estos datos se utilizan para fortalecer la seguridad o mejorar funcionalidades de nuestra aplicación, o cuando estamos legalmente obligados a retener estos datos por períodos de tiempo más largos.

Transferencia de sus Datos Personales

Se tomarán todas las medidas razonablemente necesarias para garantizar que sus datos se traten de manera segura y de acuerdo con esta Política de Privacidad y ninguna transferencia de sus Datos Personales se realizará a una organización o país a menos que haya los controles adecuados en vigor, incluida la seguridad de sus datos y otra información personal.

Eliminar sus datos personales

Usted tiene derecho a eliminar o solicitar que ayudemos a eliminar los Datos Personales que hemos recopilado sobre usted.

Nuestra aplicación le brinda la capacidad de eliminar cierta información sobre usted desde dentro.

Puede actualizar, modificar o eliminar su información en cualquier momento iniciando sesión en Su Cuenta, si tiene una, y visitando la sección de configuración de la cuenta que le permite administrar su perfil. También puede comunicarse con nosotros para solicitar acceso, corrección o eliminación de cualquier información personal que nos haya proporcionado.

Divulgación de los datos personales

Podremos divulgar Sus Datos Personales para acciones necesarias para: cumplir con una obligación legal, prevenir o investigar posibles conductas indebidas en relación con la aplicación, proteger la seguridad personal de los Usuarios, protegerse contra responsabilidades legales.

Seguridad de sus datos personales

La seguridad de Sus Datos Personales es importante para Nosotros, pero recuerde que ningún método de transmisión a través de Internet o método de almacenamiento electrónico es 100% seguro. Aunque nos esforzamos por utilizar medios comercialmente aceptables para proteger Sus Datos Personales, no podemos garantizar su seguridad absoluta.

Privacidad de menores

No recopilamos información de identificación personal de ninguna persona menor de 13 años. Si usted es padre, madre o tutor, y tiene conocimiento de que su hijo nos ha proporcionado Datos Personales, por favor contáctenos. Si nos damos cuenta de que hemos recopilado Datos Personales de cualquier persona menor de 13 años sin verificación del consentimiento parental, tomaremos medidas para eliminar esa información de nuestros servidores.

Cambios en la política de privacidad

Podemos actualizar nuestra Política de Privacidad de vez en cuando. Le notificaremos cualquier cambio publicando la nueva Política de Privacidad en esta página.

Le informaremos por correo electrónico y/o mediante un aviso prominente en nuestro Servicio, antes de que el cambio entre en vigencia y actualizaremos la fecha de *Última actualización* en la parte superior de esta Política de Privacidad.

Se le aconseja revisar esta Política de Privacidad periódicamente para cualquier cambio.

Contáctenos

Si tiene alguna pregunta sobre esta Política de Privacidad, puede contactarnos:

- Por correo electrónico: alvdela.rosa.zarzuelo@gmail.com

Apéndice D

Enlaces adicionales

- Enlace a los prototipos de la interfaz creados mediante la herramienta Figma: Interfaz Adultos, Interfaz Menores
- Enlace a los repositorios de GitHub con el código desarrollado durante la formación previa: <https://github.com/alvdela/SportApp>, <https://github.com/alvdela/ConceptosBasicos>
- Enlace al repositorio GitHub con todo el código de la aplicación:
<https://github.com/alvdela/SmartSpend>
- Enlace al APK de la aplicación:
<https://drive.google.com/drive/folders/1Stqs6IXHRg-zNMq-mzJZIikCjEbnGi3L?usp=sharing>