UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Gestión Personal: Desarrollo de una Herramienta para Finanzas y Tareas

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Salinas Delgado, Alejandro

Tutor/a: Valderas Aranda, Pedro José

CURSO ACADÉMICO: 2023/2024

Resumen

Diseño y desarrollo de una aplicación móvil siguiendo la metodología de diseño centrado en el usuario. Se utilizan tecnologías como Dart y Flutter, dando una solución holística que aborda tanto la gestión monetaria como la del tiempo para los usuarios. La aplicación ofrece características como seguimiento de gastos, planificación de actividades y notificaciones personalizadas.

Palabras clave: Aplicación Móvil, Firebase, Flutter, Dart, Tiempo, Economía

Abstract

Design and development of a mobile application following the user-centered design methodology. Using technologies such as Dart and Flutter and providing a holistic solution that addresses both money and time management for users. The app offers features like expense tracking, activity planning, and personalized notifications.

Key words: Mobile Application, Firebase, Flutter, Dart, Time, Economy

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO 1
Introducción

En el mundo acelerado de hoy, la gestión eficiente del tiempo y los recursos económicos es fundamental para el éxito personal y profesional. Sin embargo, a menudo es un desafío equilibrar nuestras responsabilidades diarias con nuestras metas financieras a largo plazo. Este Trabajo de Fin de Grado se centra en el desarrollo de una aplicación innovadora que aborda estos desafíos de manera integral.

La aplicación, que denominaremos "Gestión Personal" para los propósitos de este documento, tiene como objetivo proporcionar a los usuarios una plataforma intuitiva y fácil de usar para la gestión del tiempo y los recursos económicos. Gestión Personal combina diversas funcionalidades para la organización personal en una única solución integrada.

1.1 Objetivos Generales

El objetivo principal de este TFG es desarrollar una aplicación de gestión holística que permita a los usuarios gestionar eficazmente su tiempo y recursos económicos. Para lograr esta meta, se han establecido los siguientes objetivos generales:

- Desarrollar un Calendario Personalizado: El calendario permitirá a los usuarios programar y rastrear sus actividades diarias, semanales y mensuales. Además, el calendario estará integrado con las funciones de gestión económica de la aplicación, permitiendo a los usuarios visualizar cómo su tiempo se traduce en gastos e ingresos.
- Implementar un Rastreador de Gastos: Esta función permitirá a los usuarios registrar y categorizar sus gastos diarios. Los usuarios podrán ver resúmenes de sus gastos, lo que les ayudará a entender mejor sus hábitos de gasto y a identificar áreas potenciales de ahorro.
- Crear un Planificador Financiero: El planificador financiero ayudará a los usuarios a establecer y seguir metas financieras a largo plazo. Los usuarios podrán introducir sus gastos y metas financieras para ser capaces de seguirlas hasta alcanzar sus objetivos.

En resumen, este TFG se propone desarrollar una aplicación que ayude a los usuarios a gestionar su tiempo y dinero. A través de Gestión Personal, los usuarios podrán tomar decisiones más informadas y proactivas sobre cómo utilizan sus recursos, lo que en última instancia les permitirá alcanzar sus metas personales y financieras.

CAPÍTULO 2
Estado del arte

En este capítulo se presenta un análisis del estado del arte en el campo relevante para nuestra investigación, el de las aplicaciones de gestión monetaria y de tiempo. Este análisis permitirá contextualizar nuestro trabajo dentro del panorama actual y destacar las contribuciones previas que han influido en el desarrollo de nuestra solución.

2.1 Travelspend

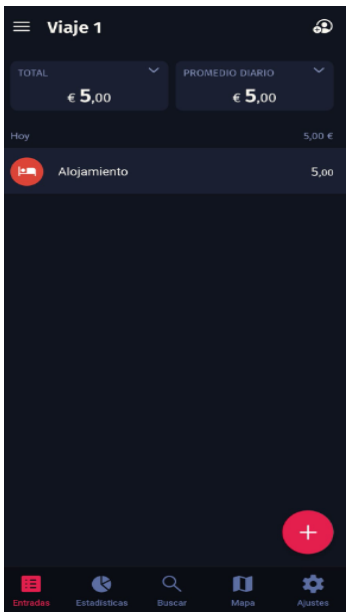


Figura 2.1: Pantalla Principal de la aplicación TravelSpend.

La aplicación mostrada en la figura 2.1 es una app freemium de gestión de gastos centrada en actividades realizadas durante los viajes. Es capaz de gestionar datos de diferente índole, esto es, diferentes tipos de gastos así como su precio. El tipo de gasto es personalizable, pero esta característica es premium y se necesita realizar un gasto en la aplicación para obtenerse.

Esta aplicación, al estar relacionada con los viajes, posee un mapa, útil para localizar los gastos y recordarlos según la localización. Además, también posee gráficas estadísticas que permiten reconocer fácilmente en que se han efectuado los gastos, dando así una versión general de todo lo que se ha gastado en el viaje y los porcentajes en cada área, ayudando a la organización de futuros viajes y a la gestión del dinero en el viaje actual.

2.2 Monefy

Figura 2.2: Pantalla Principal de la aplicación Monefy.

La aplicación mostrada en la figura 2.2 es una aplicación gratuita que permite añadir un saldo, es decir, una suma del dinero sobre el que se quiere mantener el control, y a este sumarle ingresos o gastos, asignados a diferentes categorías preestableciadas y no personalizables, como lo pueden ser regalos o taxis.

Tiene la posibilidad de mantener el recuento de lo que se posee tanto en efectivo como en la tarjeta, añadiendo una capa de complejidad a la aplicación y permitiendo control tanto de lo que se tiene en la caja de ahorros como de lo que se tiene en casa.

Posee también opción de cambio monetario, para pasar los ahorros a otro tipo de moneda, tras, por ejemplo, una mudanza.

2.3 1Money

Figura 2.3: Pantalla Principal de la aplicación 1Money.

La aplicación freemium mostrada en la figura 2.3 permite una fácil creación de presupuestos, enfocados en categorías específicas, los cuales representan tipos de gastos dentro de un ámbito determinado. Estas categorías pueden ser predefinidas o personalizadas por el usuario.

Además, permite registrar tanto ingresos como gastos asociados a dichas categorías. La aplicación también ofrece la posibilidad de vincular tarjetas bancarias para que pueda rastrear automáticamente los gastos.

Por último, tiene un apartado resumen con unos gráficos útiles para el estudio estadístico de los gastos que se van realizando, según los gastos asignados.

2.4 Spendee

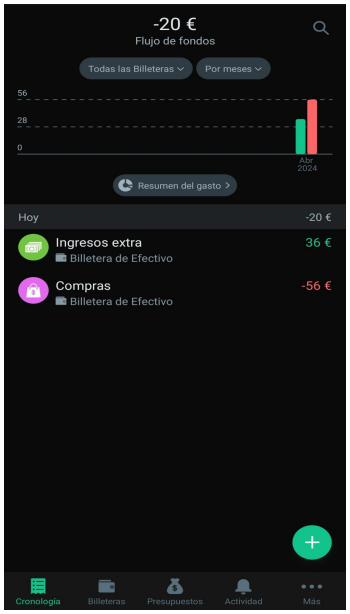


Figura 2.4: Pantalla Principal de la aplicación Spendee.

Esta aplicación, mostrada en la figura 2.4 permite la creación de un presupuesto centrado en un tipo de categoría, que son tipos de gastos centrados en un campo determinado, como pueden ser los deportes o el cine, por ejemplo, y que pueden ser predeterminados o creados por el usuario. También permite la creación de presupuestos generales donde se añade todo.

Dentro de esos presupuestos, la aplicación permite tanto añadir ingresos como gastos, permitiendo así un seguimiento ordenado, además, por meses o años, de lo que se va tanto gastando como ingresando. Los presupuestos pueden ser de diferentes duraciones, como quincenas o meses.

Puedes añadir tanto una tarjeta bancaria como una cryptowallet para que la propia aplicación monitorize los gastos, como característica premium de la aplicación.

2.5 MoneyHero

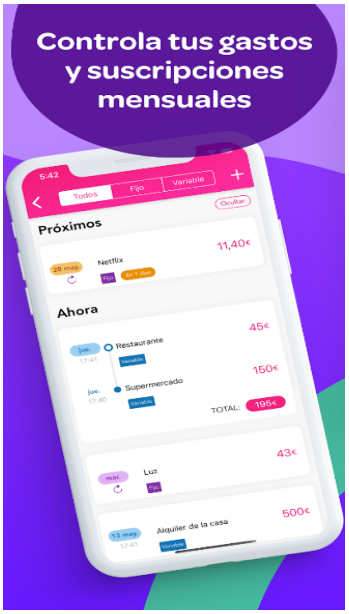


Figura 2.5: Pantalla Principal de la aplicación MoneyHero.

Esta aplicación, que aparece en la figura 2.5, es freemium. Permite planificar gastos y eventos asociados a estos, organizados según un orden cronológico. Esta aplicación también permite coordinar finanzas con tus amigos o pareja, haciéndoles partícipes de esos eventos y repartiendo gastos a posteriori. Repartir gastos es una característica premium.

Además permite analizar dónde se ha gastado el dinero con diversas y útiles gráficas y establecer objetivos de ahorro, límites a lo que se gasta en un mes.

Actualmente se encuentra desactualizada para la última versión de android, limitando las posibilidades de ser adquirida en nuevos dispositivos.

2.6 Tabla Comparativa

Tabla 2.1: Tabla comparativa de aplicaciones

Caract.	**TS**	**MF**	**1M**	**SP**	**MH**	**Gestión Personal**
Costo	F	G	F	F	F	G
Gráficos estad.	S	S	S	S	S	N
Presupuestos	N	N	S	S	N	S
Calendario	N	N	N	N	N	S
Vinc. tarj. banc.	N	N	S	S	N	N
Vinc. cryptowallet	N	N	N	S	N	N
Coord. financiación	N	N	N	N	S	N
Notificaciones	N	N	N	N	S	S

- **TS** = Travelspend
- **MF** = Monefy

- **1M** = 1Money
- **SP** = Spendee
- **MH** = MoneyHero
- **F** = Freemium
- **G** = Gratis
- **S** = Sí
- **N** = No

La aplicación desarrollada se destaca principalmente, como se puede observar en la tabla 2.1, por su calendario funcional. A diferencia de otras aplicaciones que pueden organizar los gastos y presupuestos por fechas, esta aplicación ofrece un elemento visual que permite mostrarlos de forma simple a lo largo de un mes. Además, cuenta con notificaciones para avisar sobre los eventos próximos, una característica que la mayoría de las otras aplicaciones no posee.

El enfoque holístico de esta aplicación, por tanto, es único, ya que integra tanto tareas con costos y su planificación, como los presupuestos para aportar una organización y un control sobre las propias tareas, mientras que el resto de aplicaciones no le otorgan relevancia al apartado de planificación temporal. Esto le otorga una profundidad que falta en otras aplicaciones, las cuales se centran únicamente en la planificación económica y usan métodos temporales básicos y meramente indicativos, sin la capacidad de planificar diversos elementos a lo largo del tiempo de manera detallada y sin la posibilidad de notificar cuando estos suceden.

Hay que tener en cuenta, además, que algunas de las características presentes en otras aplicaciones son premium, y no todo el mundo está dispuesto a pagar por ellas.

Como conclusión, el enfoque holístico aporta algo nuevo al estado actual del arte, dotando de profundidad temporal a la planificación económica y viceversa.

CAPÍTULO 3
Metodología

La metodología empleada en la realización de esta aplicación es el diseño centrado en el usuario, o DCU [1] por sus siglas. El enfoque que el DCU aporta es el de desarrollar un producto adecuado y personalizado para un usuario, centrando el punto de mira en quién va a usar el producto, cómo va a usarse y qué objetivos el usuario se propone.

El usuario, según este enfoque, debe ser el centro de atención a la hora de tomar decisiones de diseño. Se busca, aplicando el DCU, el diseño de una experiencia, no solo de un producto, ya que se pretende entender este producto dentro del mundo que le rodea, de su contexto. También se aspira a hacerlo de forma útil para satisfacer los requerimientos y propósitos del usuario.

El concepto de usabilidad [2] es un elemento central en el enfoque DCU, pero no se ha de confundir este concepto con el propio DCU, ya que el DCU es, en definitiva, una metodología destinada a obtener una mejor usabilidad.

El procedimiento DCU logrará cumplir sus metas si es capaz de cumplir los objetivos de sus usuarios a través de que estos obtengan una alta calidad de uso. Por lo tanto el objetivo final del DCU es satisfacer a los usuarios adaptando la tecnología y las interfaces para facilitar la consecución de sus objetivos.

El DCU es un proceso de aplicación cíclica, en el que las decisiones de diseño están dirigidas por el usuario y aquellas metas que se pretenden alcanzar mediante el uso del producto. Durante este proceso se pone el foco en la usabilidad, que se ve aumentada incrementalmente.

Podemos separar este proceso en cuatro partes, que aparecen en la figura 3.1 :

Comprensión y especificación del contexto de uso: En esta fase, se identifican las personas que utilizarán el producto, así como sus necesidades y las condiciones en las que lo usarán.

Especificación de requisitos: Aquí se establecen los objetivos que deben cumplir tanto el usuario como el proveedor del producto. Se definen los requisitos que deben satisfacerse para lograr estos objetivos.

Generación de soluciones de diseño: Esta etapa implica la creación de diversas soluciones conceptuales que aborden los requisitos especificados. Se desarrollan diferentes opciones de diseño que van desde ideas iniciales hasta una solución final.

Evaluación: Esta fase es crucial ya que se verifica si las soluciones de diseño cumplen con los requisitos establecidos y si son efectivas para el usuario. Se realizan pruebas y evaluaciones con usuarios para detectar posibles problemas de usabilidad y mejorar el diseño en consecuencia.

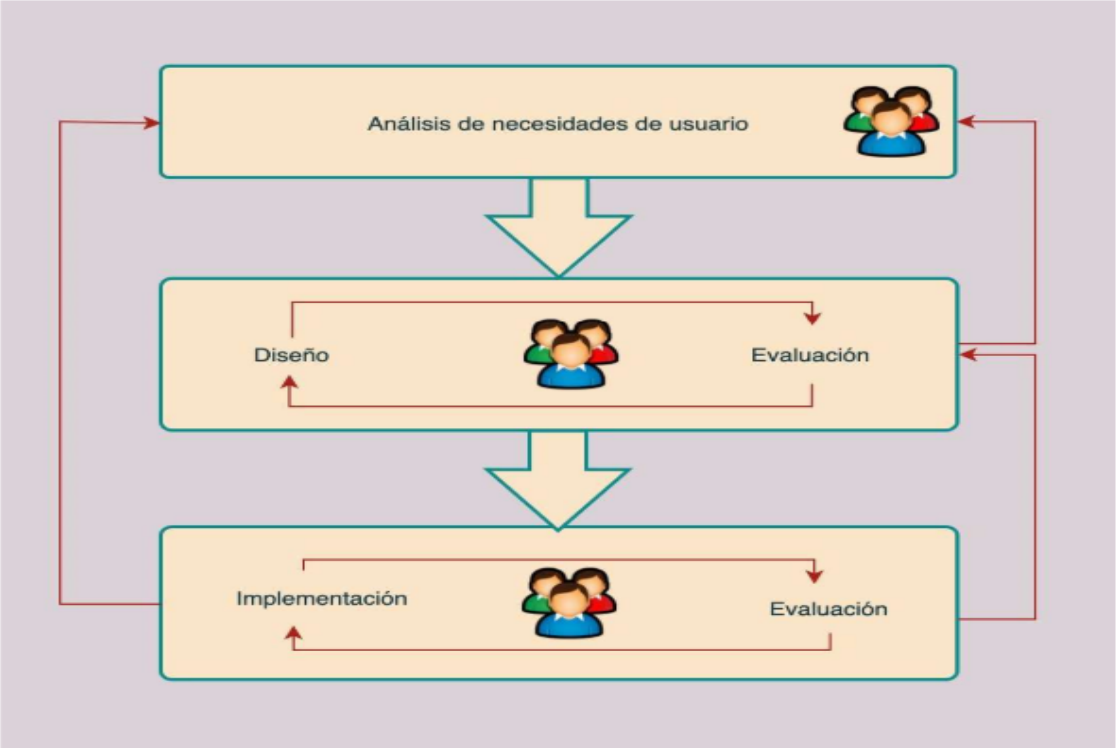


Figura 3.1: Metodología DCU

3.0.1. Comprensión y especificación del contexto de uso

En la primera parte de este proceso de aplicación cíclica, comprensión y especificación del contexto de uso, se busca centrar los esfuerzos en conocer los hábitos y las necesidades que nuestros usuarios poseen, tanto el qué como el por qué de cada uno de ellos. Para ello se pretende realizar una técnica de análisis separada en tres fases:

Investigación cualitativa

La primera fase consiste en la realización de una investigación cualitativa sobre los hábitos de los usuarios, de los cuáles se deducen sus necesidades, averiguando tanto la tarea como el porqué de su realización. En esta fase es clave la recopilación de datos a través de entrevistas, las observaciones sobre los usuarios y la retroalimentación que estos puedan aportar.

Técnica personas

La segunda fase trata de la utilización de la técnica personas, desarrollada por Alan Cooper, creador de Visual Basic y trabajador de Microsoft.

Comenzó con un prototipo de esta técnica en 1983, usando entrevistas que realizaba con grupos pequeños de personas. Desde 1995 se centró en la utilización por parte de un usuario particular del sistema desarrollado, tanto software como interfaz, en lugar de uno generalizado. La técnica se popularizó en la comunidad tecnológica a raíz de la publicación de su libro The Inmates are Running the Asylum, en 1999. El autor, en este libro, señala las características principales y buenas prácticas de la técnica que él mismo ha desarrollado. También recomienda el desarrollo de los productos para una persona arquetícipa y alejándose de la perspectiva de un experto en tecnología [3].

La técnica personas permite resumir y sintetizar la información recopilada en la fase de investigación y usarla como base para comenzar el diseño. En este momento es cuando se comienza el llamado modelado de usuario, que consiste en la creación de perfiles de usuario en base a atributos compartidos y usualmente distribuidos en rangos para mayor generalicidad, sustraidos de la investigación realizada en la primera fase. Estos perfiles de usuario son una descripción detallada del conjunto de atributos. Se utiliza principalmente para que durante el desarrollo se tenga conocimiento compartido sobre los usuarios reales, proporcionándo contexto, capacidades y objetivos a los encargados de desarrollar el producto.

El modelado de usuario da como producto a una persona, es decir, un personaje ficticio y arquetípico que personifica los objetivos y motivaciones de un grupo de personas en un individuo, una personificación que ayuda a empatizar a los desarrolladores. En esta fase el punto más importante a conocer de la persona son los objetivos que esta desea alcanzar, ya que estos serán utilizados a la hora de desarrollar el producto.

Escenarios

En la última fase del análisis de necesidades se desarrollan los escenarios, que son unas descripciones de como los personajes arquetípicos anteriormente creados con la técnica personas utilizan el producto para conseguir sus objetivos. Suelen ser varios relatos cortos en los que se desarrollan las tareas, en un ambiente concreto. Se desarrollan de más general a más específico, teniendo en cuenta que será importante contar con uno general antes siquiera de comenzar con el diseño y este irá avanzando hacia modelos más específicos conforme avance el propio diseño. La información necesaria para crear un escenario es diferente a la utilizada para la persona, ya que en este caso se utilizan las necesidades de información de los usuarios, las acciones que pueden llevar a cabo y las funcionalidades.

Con este desarrollo se pretenden identificar aspectos importantes que tengan relevancia a la hora de utilizar el producto y que son difícilmente localizables de otro modo. Además, son útiles a lo largo de todo el proceso de diseño para, por ejemplo, desarrollar las tareas en las pruebas de usabilidad. Hay que añadir que estos escenarios nos facilitan realizar hipótesis sobre las situaciones y necesidades de los usuarios, pero que en ningún caso nos permiten identificar interacciones concretas, tan solo nos muestran tanto el contexto como los objetivos a conseguir.

El objetivo de los escenarios, por lo tanto, es conocer tanto para quíen se está diseñando como lo que quiere el usuario del producto y cómo piensa usarlo para alcanzar sus objetivos. También es de relevancia el porqué este se ha decantado por nuestro producto y no por otro. A diferencia de la ingeniería del software, dónde se usan los casos de uso para modelar los requisitos y describir las funcionalidades del sistema, en este caso usaremos los escenarios como un recurso para ayudar a entender al usuario y el uso que hará del sistema. Gracias a estos escenarios pretendémos, como fin último, mejorar la usabilidad y conseguir una buena experiencia de usuario.

3.0.2. Especificación de requisitos

En la segunda parte de este proceso, especificación de requisitos, usaremos los escenarios desarrollados en la anterior parte para definir con mayor precisión los requisitos del sistema. Los escenarios desarrollados en la parte anterior nos proporcionan casos concretos de uso del sistema, lo que nos permite identificar las diferentes funciones y características que deben ser implementadas.

Utilizaremos estos escenarios como base para elaborar una lista detallada de requisitos funcionales y no funcionales que el sistema debe cumplir. Esto incluirá detalles como la funcionalidad específica que debe ser proporcionada, los comportamientos esperados en diversas situaciones, las restricciones de rendimiento, seguridad y usabilidad, entre otros aspectos relevantes para el desarrollo del sistema.

Además, los escenarios nos ayudarán a validar y verificar los requisitos a medida que avancemos en el proceso de especificación, asegurando que el sistema cumpla con las expectativas de los usuarios.

3.0.3. Generación de soluciones de diseño

Para la tercera parte del proceso, generación de soluciones de diseño, utilizaremos los requisitos especificados en la segunda parte como guía para proponer diferentes soluciones de diseño que cumplan con dichos requisitos. En esta etapa, se explorarán diversas alternativas arquitectónicas, tecnológicas y de implementación para el sistema, considerando factores como la eficiencia, la escalabilidad, la mantenibilidad y la viabilidad técnica y económica.

Se pueden emplear técnicas como el diseño conceptual, diagramas de flujo, modelos de datos, prototipado y simulaciones para visualizar y evaluar las diferentes soluciones propuestas. Además, es importante tener en cuenta las restricciones y limitaciones del proyecto, así como las necesidades y expectativas de los usuarios finales.

El objetivo principal de esta fase es identificar la mejor solución de diseño que satisfaga de manera óptima los requisitos del sistema, maximizando la calidad y minimizando los riesgos asociados con su implementación. Es fundamental involucrar a todas las partes interesadas relevantes en este proceso para garantizar que las soluciones propuestas sean adecuadas y viables desde diferentes perspectivas.

3.0.4. Evaluación

Por último, en la fase de evaluación, se lleva a cabo un examen meticuloso de las diversas soluciones de diseño propuestas en la etapa anterior. Esta fase reviste una importancia crucial, ya que se trata de verificar no solo si las soluciones cumplen con los requisitos establecidos, sino también si son efectivas y satisfactorias para el usuario final. Se emplean pruebas exhaustivas y evaluaciones directas con usuarios reales para identificar posibles problemas de usabilidad y mejorar el diseño en consecuencia.

Durante este proceso, se realizan pruebas rigurosas y se recopila retroalimentación detallada de los usuarios para comprender mejor sus necesidades y preferencias. Se utilizan técnicas como la observación directa, encuestas y entrevistas para capturar una amplia gama de opiniones y experiencias. Este enfoque centrado en el usuario garantiza que la solución final no solo sea funcional, sino también intuitiva y fácil de usar para aquellos que la emplearán en la práctica.

La retroalimentación obtenida durante estas pruebas es esencial para realizar ajustes finales en el diseño y garantizar que la solución adoptada cumpla con los estándares de calidad esperados. Se trata de un proceso iterativo que puede implicar múltiples rondas de evaluación y refinamiento antes de llegar a una solución final. Al finalizar esta fase, se tiene la confianza de haber seleccionado la opción de diseño más adecuada, con un alto grado de certeza de que cumplirá con las necesidades y expectativas del usuario final.

CAPÍTULO 4
Análisis de necesidades

En este capítulo se pretende realizar un análisis, a partir de la recogida de datos obtenidos a través de un formulario, de las necesidades de las personas. A raiz del estudio de estas respuestas se planteará una persona arquetípica, tomando como base las diferentes preguntas que han contestado los encuestados.

Partiendo de las contestaciones se generarán diversos parámetros como sus motivaciones, sus frustraciones y su frase modelo, todo ello para intentar dotar de realismo a los objetivos de la persona, que son los que promueven finalmente el desarrollo de la aplicación.

Por último se generaran escenarios a cumplir por la persona, basados en unas historias cortas en las que se enfatiza el qué hace para lograr sus objetivos, no el cómo lo hace, permitiendo que desarrollemos un producto a su medida para que logre los objetivos que se ha fijado.

4.1 Resumen de los cuestionarios hechos y resultados

Se ha realizado un cuestionario con una sección demográfica, una tecnológica y una centrada en los apartados concretos de la aplicación.

Este cuestionario cuenta con una batería de 44 preguntas a responder por los encuestados, las cuales se encuentran en el anexo A.

Las respuestas nos han permitido abordar la creación de la persona. Algunos detalles que podemos observar a raíz de ciertas preguntas son, por ejemplo, en el apartado biográfico, una mujer casada y con hijos, fruto de las respuestas a esta sección de las encuestas realizadas. También podemos observar gracias a este apartado que tiene un trabajo de maestra a tiempo completo. Esto surge de las respuestas tanto a la pregunta de la figura 4.1 como a la de la 4.2.

Los pasatiempos de la persona surgen de la pregunta diréctamente relacionada con estos, mostrada en la figura 4.3, y de que pasatiempos se han escogido mayoritariamente.

Tanto las motivaciones como los objetivos han sido deducidos del resto de las preguntas del test, el apartado principal. Una pregunta significativa podría ser la 4.4 Debido a que muestra que la mayor parte de las personas realizan, aunque no de forma frecuente, una planificación mensual.

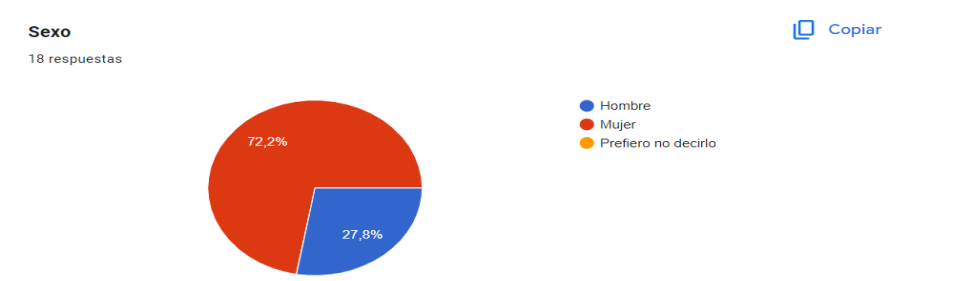


Figura 4.1: Sexo de los encuestados

Figura 4.2: Trabajo de los entrevistados

Figura 4.3: Pasatiempos de los entrevistados

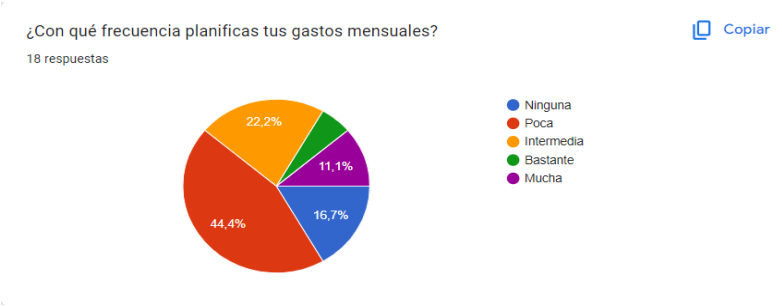


Figura 4.4: Pregunta considerada significativa

4.2 Definición de Persona

- Frase: "¡Con pasión y dedicación, construyo un presente lleno de aprendizaje y amor!"

Figura 4.5: Persona: Luisa Martín Ochoa

- Nombre: Luisa Martín Ochoa
- Edad: 43 años
- Profesión: Profesora de matemáticas
- Estado Civil: Casada
- Biografía: Luisa vive con su pareja desde hace 8 años y tiene 2 hijos. Logró obtener su plaza de fija de profesora hace 3 meses. Está motivada a la hora de enseñar a sus alumnos, ya que les ayuda a aprender y a formarse para un futuro.

  Es muy activa, sus hijos tienen actividades extraescolares a las que ella y su pareja les llevan y también realiza senderismo el fin de semana. Además, le encanta el cine y la música.

  Usa habitualmente Whatsapp y Instagram en su dispositivo móvil para chatear con sus conocidos y compartir sus imágenes.
- Personalidad: Diligente/Dedicada/Altruista/Involucrada
- Objetivos/Motivaciones:
  - No ir justa de tiempo a la hora de dedicarlo tanto a sus hijos como a sus pasatiempos.

    - Llevar a tiempo a sus hijos a cada una de sus actividades, así como saber a que hora comienzan las sesiones de cine en las que está interesada.
    - Poder planificar sus gastos con tiempo para organizar sus ahorros

- Frustaciones:

    - Llegar tarde a sus citas.
    - No llegar a fin de mes debido a su excesivo gasto en actividades.
    - No tener tiempo para el ocio y sus pasatiempos.

4.3 Escenarios

4.3.1. Escenario uno

Es viernes por la tarde y Luisa está disfrutando de un momento tranquilo en casa después de una semana agitada. Mientras sus hijos juegan en la sala de estar, Luisa saca su teléfono móvil y abre la aplicación que ha estado utilizando para organizar su vida.

En la pantalla, ve un resumen de todas las actividades programadas para la próxima semana. Desde sus clases de matemáticas hasta las sesiones de estudio personal, pasando por las actividades extracurriculares de sus hijos, todo está meticulosamente organizado en fechas y horas específicas.

Sin embargo, Luisa recuerda que necesita agregar una cita importante con el director de la escuela de su hijo mayor para discutir su progreso académico. Con unos pocos toques en la pantalla, Luisa encuentra un hueco disponible en su calendario y agrega la cita, asegurándose de no superponerse con ningún otro compromiso.

Después de completar la programación, Luisa decide activar algunas alertas para recordarle los eventos más importantes. Configura notificaciones para las reuniones de padres y maestros, así como para los exámenes y proyectos importantes que tiene pendientes. De esta manera, se asegura de no olvidar ninguna tarea crucial en medio de su ajetreada vida como profesora y madre.

Con la tranquilidad de tener todo bajo control, Luisa vuelve a sumergirse en su momento de relajación, lista para disfrutar del fin de semana con su familia sabiendo que la próxima semana estará perfectamente planificada y organizada.

4.3.2. Escenario dos

Es principios de mes y Luisa se sienta en su sofá con su teléfono en mano, lista para abordar una de sus principales preocupaciones: su presupuesto familiar. Consciente de que sus actividades extraescolares y salidas al cine pueden representar un desafío para sus finanzas, decide aprovechar una aplicación de gestión financiera para organizarse.

Abre la aplicación y comienza a revisar sus gastos del mes anterior, notando algunas áreas en las que gastó más de lo planeado. Con esta información en mente, se propone establecer un presupuesto más estricto para el próximo mes, asignando cantidades específicas a categorías como entretenimiento, alimentación y actividades extracurriculares de los niños.

Luisa utiliza la función de programación de la aplicación para establecer recordatorios regulares sobre sus objetivos de gasto mensuales. Configura alertas que le avisen cuando esté cerca de alcanzar el límite establecido para cada categoría, lo que le permitirá ajustar sus gastos si es necesario para evitar sobrepasar su presupuesto.

Con su nuevo plan financiero en marcha, Luisa se siente más segura para controlar sus gastos y planificar sus ahorros. Ahora puede disfrutar de sus actividades y pasatiempos sin preocuparse por excederse en sus gastos, sabiendo que está trabajando activamente hacia un futuro financiero más estable para ella y su familia.

4.3.3. Justificación

Justificación del escenario 1: Organización y Planificación de Tiempo

Este escenario aborda la necesidad de Luisa de poder planificar sus actividades con anticipación para evitar conflictos y olvidos. La creación de este escenario permite mostrar cómo una aplicación de calendario puede ayudar a Luisa a organizar sus compromisos y asegurarse de que no se le escape ninguna tarea importante.

Al utilizar una aplicación de calendario, Luisa puede tener una visión clara de todas sus actividades programadas, desde sus clases de matemáticas hasta las reuniones de padres y maestros de sus hijos. Al configurar alertas y recordatorios para eventos importantes, puede garantizar que no se le pase por alto ninguna tarea crucial en medio de su ajetreada vida como profesora y madre.

Justificación del escenario 2: Planificación Financiera

Este escenario aborda la preocupación de Luisa por no llegar a fin de mes debido a su excesivo gasto en actividades. La creación de este escenario permite mostrar cómo una aplicación de gestión financiera puede ayudar a Luisa a controlar sus gastos, establecer límites presupuestarios y planificar sus ahorros para alcanzar sus metas financieras a largo plazo.

Al establecer un presupuesto y asignar cantidades específicas a diferentes categorías de gastos, Luisa puede tener una visión clara de su situación financiera y tomar decisiones informadas sobre cómo distribuir sus recursos. Además, al configurar alertas y recordatorios, puede mantenerse al tanto de su progreso y ajustar su comportamiento financiero según sea necesario para cumplir con sus objetivos.

Conclusión

En resumen, ambos escenarios ofrecen soluciones tecnológicas prácticas y realistas para abordar las preocupaciones y objetivos específicos de Luisa en cuanto a la gestión de su tiempo y recursos financieros.

CAPÍTULO 5
Análisis Conceptual y Diseño

En el subsiguiente capítulo, se abordará detalladamente la creación del diagrama de clases [4] de la aplicación, teniendo en cuenta los escenarios establecidos en los capítulos anteriores. Este diagrama será fundamental para definir las entidades principales, sus atributos, métodos y las relaciones entre ellas, asegurando una estructura coherente y lógica del sistema.

Además, se elaborará el modelo de la base de datos [5] que será utilizado por el servidor de bases de datos. Este modelo incluirá la definición de tablas, campos y relaciones necesarias para garantizar la integridad y eficiencia del almacenamiento de datos.

Finalmente, se presentarán y explicarán diversos bocetos sobre las interfaces de la aplicación. Estos bocetos servirán como guía para el diseño final de las interfaces de usuario, asegurando que todas las funcionalidades importantes estén presentes y accesibles.

5.1 Diagrama de clases

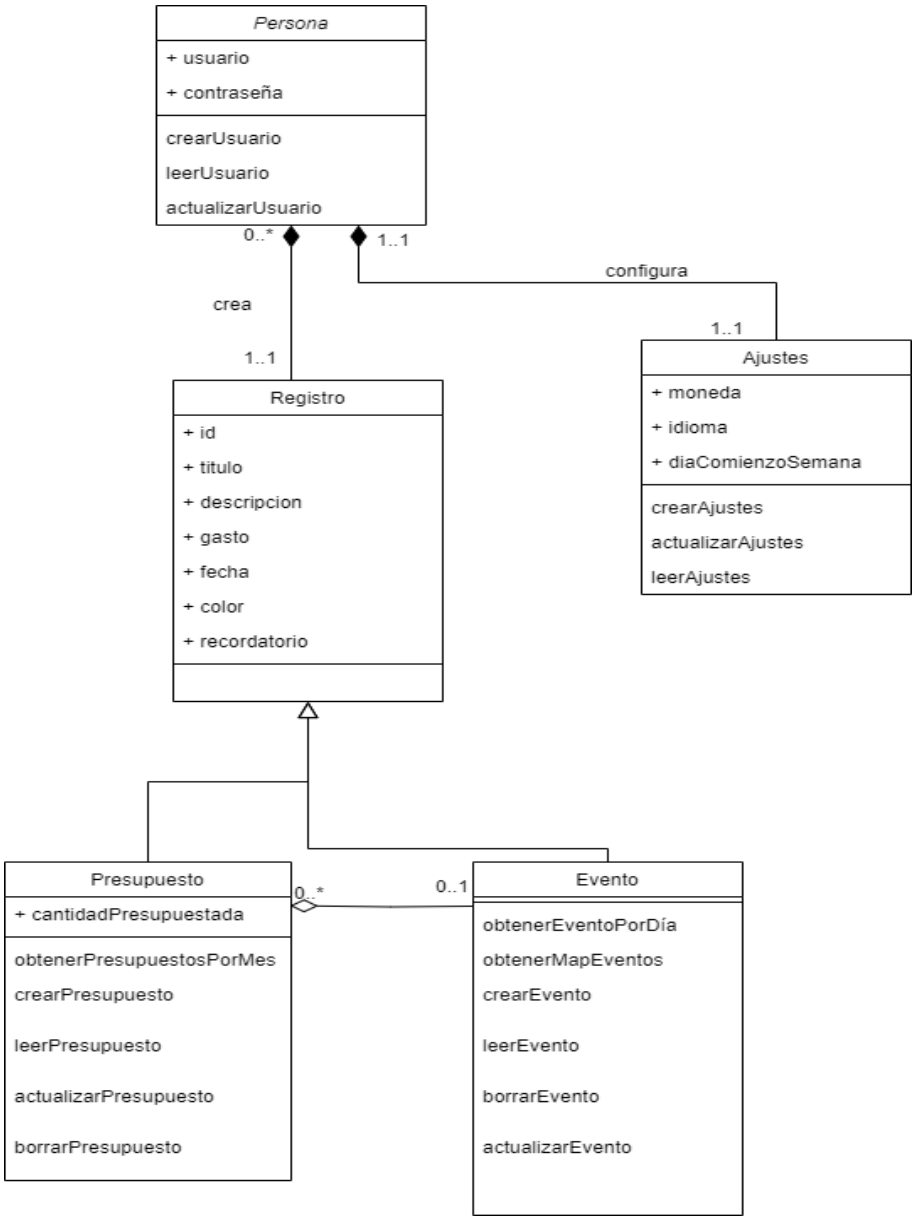


Figura 5.1: Diagrama de clases

Este diagrama de clases, figura 5.1 cuenta con la clase persona, extraída directamente de Luisa, nuestra persona, que cuenta con un usuario y una contraseña para poder acceder al sistema. Esta persona crea registros, que son elementos que usa para planificar su dia a dia. Para organizarse cuenta con tanto presupuestos como eventos, que son los elementos que esta registra en el sistema. Un evento se relaciona por agregación a un presupuesto, ya que un presupuesto agrupa eventos para calcularse, es decir, agregamos eventos para conformar un presupuesto, pero este tiene sentido sin los eventos, ya que indica que no se ha gastado nada de momento. Por último cuenta con unos ajustes que le permiten adaptar a sus gustos o necesidades la aplicación. Tanto los registros como los ajustes se relacionan por composición con la persona, ya que estos no tienen sentido que existan si no están relacionados con esta.

Los diferentes campos hacen referencia al grado de personalización que se ha querido añadir a los registros para que estos sean identificables y del agrado del usuario, además de las utilidades con las que contará la aplicación, como las notificaciones.

Como conclusión, este diagrama de clases cubre los escenarios de la persona, permitiendo crear los elementos organizativos vistos en los escenarios y añadiendo, además, cierto nivel de personalización para que esta se encuentre cómoda con el uso de la aplicación.

5.2 Modelo de la BD que use el servidor

El modelo de la base de datos que usa el servidor, figura 5.2, cuenta con cuatro tablas. Estas nos permiten modelar los diferentes objetos que usa la aplicación para que, a la hora de implementar la base de datos, estos puedan adquirir persistencia en el sistema.

Estos objetos son personas, ajustes, eventos y presupuestos, similar a los elementos vistos en el diagrama de clases. Se ha decidido no incluir el registro debido a que las operaciones generalmente no incluyen elementos de ambas tablas, eventos y presupuestos, sino de una u otra, por lo que se ha optado por mantenerlas por separado para así poder realizar consultas más simples y eficientes.

Los campos obligatorios se han mantenido como no nulos y las claves se han marcado con una pequeña llave al lado del nombre asignado, para mantener la coherencia.

En definitiva, esta base de datos nos permitirá dar una persistencia, organización, coherencia y flexibilidad a los datos que deseemos almacenar, relacionados con los escenarios propuestos, permitiéndonos un tratamiento eficaz de los mismos.

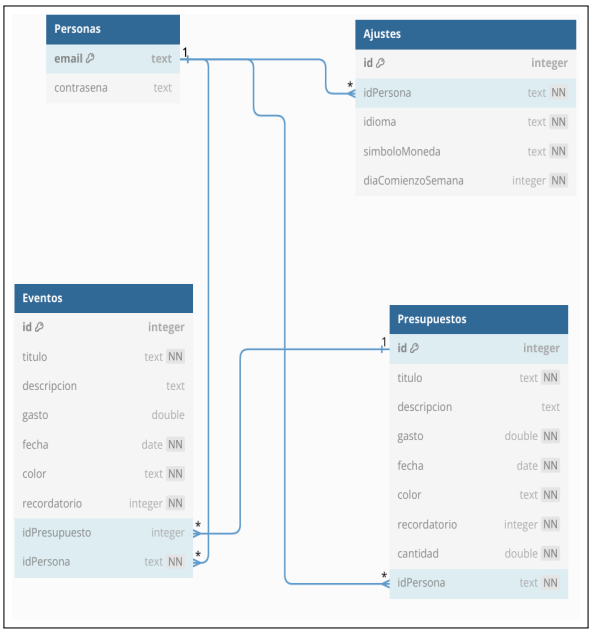


Figura 5.2: Modelo de la base de datos

5.3 Bocetos de las interfaces de la app

Estos bocetos están realizados con el sitio web Figma, el cuál es una potente herramienta en línea que permite la creación y edición de gráficos vectoriales y es un excelente

diseñador de prototipos altamente personalizable. Este cuenta tanto con plantillas prediseñadas por la comunidad como con una gran cantidad de opciones para adaptar a las necesidades propias los prototipos que se realizan.

Figura 5.3: Boceto del formulario de crear la cuenta

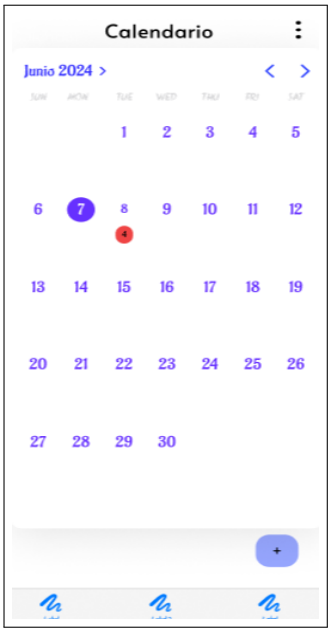


Figura 5.4: Boceto del calendario

EL boceto 5.3 nos permite crear con un simple formulario una cuenta y acceder al login, de apariencia similar, en caso de que el usuario ya posea una cuenta. Este boceto cuenta con un título, tres campos de texto, dos de ellos ocultos para la contraseña, un botón para iniciar el registro y un enlace a la página de inicio de sesión.

En el boceto 5.4 podemos observar la ventana principal de la aplicación, un calendario que posee un evento marcado en rojo, además también se puede ver un menú de navegación abajo, un botón flotante con un más, y un menú desplegable con tres puntos arriba. Estos diversos elementos permiten la navegación por todas las ventanas de la aplicación.

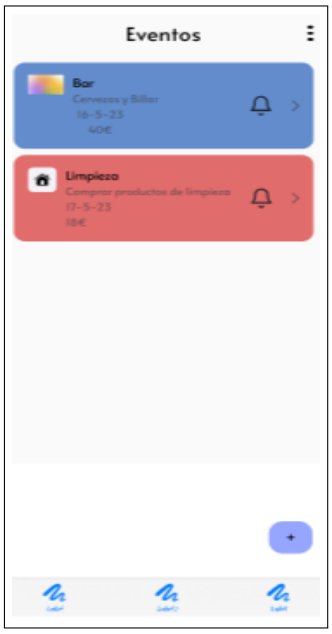


Figura 5.5: Boceto de la lista de eventos

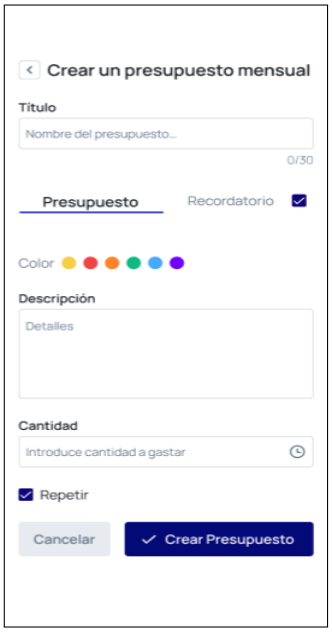


Figura 5.6: Boceto del formulario de la creación de presupuesto

En la figura 5.5 se puede ver como el título y el calendario ha cambiado, pero seguimos en la ventana principal, con los mismos elementos de navegación. En lugar del calendario tenemos una lista de eventos con las diversas características que los definen. En el caso de los presupuestos nos encontraríamos con un caso parecido, aunque ciertas características de cada uno de los elementos de la lista cambiarían para adaptarse al nuevo tipo de objeto.

Para continuar, en la figura 5.6 encontramos un formulario con los diversos campos que modelan un presupuesto. Este formulario se envía con el botón inferior. En este caso la ventana es diferente a la principal y permite la navegación de vuelta a esta sin la necesidad de enviar el formulario con tanto un botón inferior como uno situado al lado de título de la pantalla. El formulario para la conformación de eventos sería similar a este.

Figura 5.7: Boceto del formulario de los ajustes

Por último nos encontramos con los ajustes, una lista con varios formularios que permite la personalización de la aplicación en una ventana diferente de la principal y similar a la de la creación tanto de eventos como de presupuestos.

CAPÍTULO 6
Desarrollo de la solución

En este capítulo se profundiza en el proceso integral de desarrollo de la solución, comenzando por la concepción y diseño detallado de la arquitectura empleada en la creación de la aplicación. Se exploran las decisiones arquitectónicas clave que guian la estructura del sistema, asegurando su escalabilidad, eficiencia y mantenibilidad a largo plazo. Además, se analizan las metodologías y mejores prácticas adoptadas para garantizar un desarrollo coherente y eficaz.

Se dedica especial atención a las herramientas específicas utilizadas en el proceso de desarrollo. Desde los frameworks hasta las librerías esenciales. Estos elementos no solo facilitan la implementación de funcionalidades complejas, sino que también aseguran una base técnica robusta y actualizada.

Para ilustrar la implementación práctica de la solución, se presentan ejemplos concretos del código que conforma el sistema. Estos ejemplos no solo demuestran la aplicación de conceptos teóricos en situaciones reales, sino que también proporcionan una visión detallada de cómo se abordan y resuelven los desafíos específicos de la implementación. Cada fragmento de código seleccionado destaca aspectos clave del diseño y la funcionalidad del sistema, ofreciendo puntos de vista valiosos sobre su utilidad y rendimiento.

6.1 Arquitectura

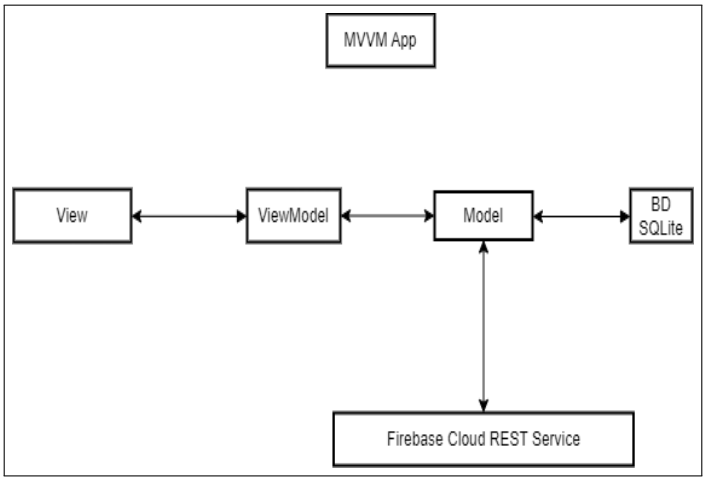


Figura 6.1: Esquema con la arquitectura

La arquitectura del sistema, mostrada en la figura 6.1, es fundamental para entender cómo los diversos elementos que lo conforman interactúan entre sí. Una buena arquitectura facilita la comprensión de la aplicación, la incorporación de nuevos elementos y el mantenimiento a largo plazo. Los patrones arquitectónicos definen las relaciones entre las capas de datos, negocio y presentación de las aplicaciones.

Esta aplicación está conformada por una aplicación móvil, un servidor de base de datos y un servicio web externo. En este caso, la aplicación móvil sigue el patrón Modelo-vista-modelo de vista (MVVM, por sus siglas en inglés).

6.1.1. Aplicación cliente

La aplicación cliente usa el patrón MVVM [6], que busca separar el desarrollo de la vista del modelo, delegando responsabilidades al intermediario, el modelo de vista. Esto permite una clara separación de la lógica de la interfaz de usuario.

- Vista: La interfaz de usuario, compuesta por widgets (elementos como textos, botones y listas) con los cuales el usuario interactúa. Estos widgets están suscritos a su modelo de vista, adaptándose y actualizándose según los cambios en este.
- Modelo de Vista: Este componente maneja las comunicaciones con la capa de modelo y controla los datos presentados en la vista. Permite diseñar la vista de forma declarativa y maneja el enlace de datos para asegurar que los cambios realizados en la aplicación se reflejen en el modelo y viceversa. En definitiva, el modelo de vista actúa como un intermediario entre la vista y el modelo.
- Modelo: Representa la capa de datos y negocio de la aplicación. Se comunica con las fuentes de datos, como una API REST [7] o una base de datos, y es responsable de otorgar persistencia a los datos, asegurando que estén disponibles para el modelo de vista y, por ende, para la vista. Maneja información en formato JSON [8] con metodos toJSON o fromJSON para trasladar los objetos a este formato o pasarlos de esto formato al propio modelo. Se usa el patrón singleton [9] para manejar las conexiones con la base de datos y mantenerlas abiertas en toda la aplicación.

Interacción entre Componentes

La interfaz (vista) está suscrita al modelo de vista para recibir datos y enviar actualizaciones. El modelo de vista obtiene datos del modelo y gestiona los cambios realizados por el usuario, asegurando la persistencia a través del modelo. Este flujo de datos y responsabilidades asegura que la lógica de negocio y la interfaz de usuario permanezcan separadas, facilitando el mantenimiento y la escalabilidad de la aplicación.

Librería de idiomas

Esta aplicación usa la librería l10n de flutter, capaz de adaptar textos a diversos idiomas y formatos. Esta librería lee estos datos de archivos de idiomas configurados como JSON claves-valor, con una clave común a todos los archivos y un valor que depende del archivo, ya que cada uno de estos archivos guarda un idioma.

6.1.2. Base de datos

La capa de modelo interactúa con una base de datos que se instala de forma local en el dispositivo en el que se despliega la aplicación. Esta base de datos utiliza una solución

ligera y embebida que permite gestionar datos de manera eficiente sin necesidad de un servidor de bases de datos independiente.

SQLite [10] es una biblioteca ampliamente distribuida que proporciona una base de datos relacional. Se encuentra embebida en el mismo proceso de la aplicación, mejorando el rendimiento, además es sencilla de implementar.

La aplicación accede a la base de datos mediante peticiones en lenguaje SQL. Esto incluye las operaciones CRUD básicas. La comunicación con SQLite se realiza a través de funciones de la propia librería que facilita estas interacciones, permitiendo a los desarrolladores ejecutar comandos SQL tanto usando funciones preestablecidas, como directamente usando código sql.

La misma librería se encarga de gestionar la seguridad, ya que previene con las funciones, por ejemplo, ataques de inyección, si estas se configuran de forma correcta, usando parámetros en lugar de texto directo.

Esta solución a la base de datos es, por lo tanto, una solución sencilla, potente y segura, que permite sencillez tanto de implementación como de mantenimiento, es rápida a la hora de ofrecer servicios y nos otorga seguridad ante inyecciones peligrosas en el código sql que utiliza al ser utilizada.

6.1.3. Servicio web externo

Para gestionar la creación de cuentas y permitir diversos métodos de registro se ha optado por implementar Firebase Auth [11], una API RESTFUL de google que permite gestionar tokens de identificación y tiene compatibilidad con OAuth 2.0, que permite usar otros proveedores como google o facebook para el registro.

Esta API recibe peticiones HTTP de la aplicación, gestionadas con una librería, tales como reestablecer contraseña olvidada o registrarse. Ante un registro o login exitoso devuelve un token de identidad en formato JSON, que permite usar otros servicios, como el de mantener la sesión activa a un usuario.

6.2 Contexto tecnológico

6.2.1. Dart

Lenguaje de programación multiparadigmático: funcional, imperativo, orientado a objetos y reflectivo. Soporta interfaces, clases abstractas, mixins y inferencia de tipos. Desarrollado por Google, permite desarrollar tanto aplicaciones web como aplicaciones móbiles, servidores y aplicaciones de escritorio. [12]

Figura 6.2: Logo de Dart

6.2.2. Flutter

Kit de desarrollo software de código abierto creado, al igual que Dart, por Google. Soporta diversas plataformas como lo son iOS, Android, Linux o Windows. Consta tanto de un sistema de renderizado como de un lenguaje propio para elaborar interfaces. Cuando una aplicación que utiliza flutter se compila, tanto el lenguaje como el sistema de renderizado se incluyen comprimidos, permitiendo el despliegue en múltiples plataformas de forma más simple que en otros casos donde se depende de un motor de renderizado local al dispositivo, pudiendo causar incompatibilidades. [13]

Figura 6.3: Logo de Flutter

6.2.3. Firebase

Conjunto de servicios de backend en la nube, proveído por google y gestionado desde su página web oficial. Permite el uso de servicios a través de APIs Restful tales como un sistema de bases de datos en la nube y control de inicio de sesión. Es modular, permitiendo usar solo aquellos elementos del interés del usuario.

Figura 6.4: Logo de Firebase

6.2.4. Visual Studio Code

Editor de código fuente usado para el desarrollo de este proyecto, altamente personalizable y desarrollado por Microsoft. Algunas de sus funciones son que permite debugging y tiene embebido control del código a través de git. Sus características principales son el resaltado de sintaxis, y el autocompletado y refactorizacion de código. Además, se puede personalizar con infinidad de extensiones creadas por los usuarios para añadir características nuevas, como la creación de un proyeto dart-flutter, en este caso, o potenciar las ya existentes, como una mayor personalización del resaltado de sintaxis.

6.2.5. GitHub

Repositorio de almacenamiento de proyectos propiedad de Microsoft que usa el sistema de control de versiones Git. Usado para monitorizar cambios, crear nuevas funcionalidades con seguridad y mantener el proyecto amparado en la nube.

Figura 6.5: Logo de Visual Studio Code

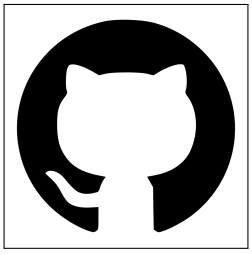


Figura 6.6: Logo de Github

6.3 Ejemplos de código

A continuación se aportarán diversos ejemplos representativos de códigos de las diferentes partes que conforman el sistema.

Interfaces

```
class Calendario extends StatelessWidget {
  final DateTime day;

  const Calendario({super.key, required this.day});

  @override
  Widget build(BuildContext context) {

    return ChangeNotifierProvider(
      create: (context) => CalendarioViewModel(context, day),
      child: Consumer2<CalendarioViewModel, LanguageProvider>(

        builder: (context, viewModel, languageNotifier, child) {

          SchedulerBinding.instance.addPostFrameCallback((_) {
            viewModel.updateLocale(languageNotifier.locale!.languageCode);
          });

          return Scaffold(
            body: Column(
              children: [
                TableCalendar<Evento>(
                  locale: viewModel.locale,
```

Figura 6.7: Ejemplo del view del calendario

En este apartado se explicará el código desarrollado para la elaboración de la interfaz del calendario de acuerdo al modelo VMMV.

Esta interfaz es un widget sin estado, es decir, que no se altera durante su ejecución, sino que requerirá ser reconstruido cada vez que se quiera alterar, para lo que utilizaremos un consumidor de eventos. Requiere, además, de un DateTime día para construirse y posee solo un método sobreescrito de la clase StatelessWidget, build, que requiere de un contexto de ejecución, que representa la parte de la interfaz que constituye este widget.

El método build, método principal y único de la vista, posee en su interior el núcleo de la interfaz, devuelve un ChangeNotifierProvider, que crea a su vista-modelo y se pone a consumir tanto los cambios en la vista-modelo como los cambios en ajustes. Cualquier cambio en uno de estos ámbitos reconstruirá la interfaz llamando a lo que hay dentro del builder del changeNotifierProvider, con los cambios ya aplicados, debido a que es un StatelessWidget y no puede ser modificado sin reconstruirse. Los cambios realizados en ajustes se notificarán a la vista-modelo mediante un PostFrameCall, que se utiliza para evitar que intente la actualización antes de existir la vista-modelo.

Por último devuelve la interfaz propiamente dicha, un Scaffold, una plantilla que permite varios hijos que se ordenan en su interior. En este caso el Scaffold posee solo una Column y en esta un child, el calendario. Además, el Scaffold permite la colocación de un botón flotante en su interior a través de una serie de atributos que se refieren tanto al widget botón como a su localización dentro del Scaffold, característica que se ha aprovechado para crear el botón de la creación de eventos.

En este caso varias características de la interfaz son sustraidas de la vista-modelo, como el idioma o la lista de eventos que han de mostrarse como puntos en el calendario.

En resumen, el código visto en la figura 6.7 implementa una interfaz de calendario con un modelo VMMV. Esta interfaz es un widget sin estado que se reconstruye gracias a los listeners. Utiliza un ChangeNotifierProvider para manejar los cambios y un Scaffold para la interfaz, que incluye un botón flotante para crear eventos. Las características de la interfaz y los datos se extraen de la vista-modelo.

6.3.1. Vista-Modelo

```dart
class ListaPresupuestosViewModel with ChangeNotifier {

  Future<List<Presupuesto>>? _presupuestosFuture;
  String simbolo = '';
  Icon? icono;
  final DateTime day;

  ListaPresupuestosViewModel(this.day) {
    _presupuestosFuture = leerPresupuestos(day);
    leerAjustes();
  }

  Future<List<Presupuesto>> leerPresupuestos(DateTime day) async {
    try {
      return await conexionDB.instance.getPresupuestosByMonth(day);
    } catch (e) {
      return [];
    }
  }

  void actualizarPresupuestos() {
    _presupuestosFuture = leerPresupuestos(day);
    notifyListeners();
  }
```

Figura 6.8: Ejemplo de vista-modelo de los presupuestos

En este apartado se describirá la vista-modelo de la lista de presupuestos de la aplicación, que se encarga de gestionar el tráfico de datos entre el modelo y la interfaz.

Las vistas-modelo see encargan tanto de presentar los datos en primera instancia como de recibir los cambios que surgen en la interfaz. También se encargan de otorgar persistancia a los cambios que recibe desde la interfaz enviándolos a la base de datos.

Esta clase tiene el mixin ChangeNotifier, que permite añadir el código perteneciente a esta clase directamente a la clase actual.

Como se puede apreciar en el ejemplo concreto de la figura 6.8, la clase lista se construye con un día, mediante el cual lee los presupuestos asignados al més al que pertenece a través de una función asíncrona, leerPresupuestos, que espera a poder realizar una conexión con la base de datos. Tras recibir la conexión, los presenta en la lista futura, una suerte de promesa, _presupuestosFuture, que es la lista que está a su vez presentada en la vista. Una vez preparada la lista, avisa a su listener de que tiene cambios que aplicar respecto a los presupuestos y este reconstruirá su builder, con los cambios ya aplicados, es decir, mostrando todos los presupuestos leidos que aparecen en la lista futura.

El proceso para gestionar cambios y dotarlos de persistencia es similar, recibe una llamada desde la vista, hace el cambio en la base de datos mediante una función asíncrona y llama a actualizarPresupuestos, para que la vista se reconstruya, esta vez ya con el cambio aplicado.

En suma, esta parte del código se encarga tanto de recibir los datos de la base de datos para posteriormente construir el widget inicialmente como de gestionar los cambios que provengan de la interfaz, presentándolos a la base de datos y reconstruyendo la propia interfaz si este cambio le afecta en algún modo.

6.3.2. Modelo

```
Map<String, Object?> toJson() => {
    CamposPresupuesto.id: id,
    CamposPresupuesto.titulo: titulo,
    CamposPresupuesto.descripcion: descripcion,
    CamposPresupuesto.gasto: gasto,
    CamposPresupuesto.fecha: fecha.toIso8601String(),
    CamposPresupuesto.color: color.value.toString(),
    CamposPresupuesto.recordatorio: recordatorio ? 1 : 0,
    CamposPresupuesto.cantidad: cantidad,
    CamposPresupuesto.idPersona: idPersona,
};


static Presupuesto fromJson(Map<String, Object?> json) => Presupuesto(
    id: json[CamposPresupuesto.id] as int?,
    titulo: json[CamposPresupuesto.titulo] as String,
    descripcion: json[CamposPresupuesto.descripcion] != null ? json[CamposPresupuesto.descripcion] as String: '',
    gasto: json[CamposPresupuesto.gasto] as double?,
    fecha: DateTime.parse(json[CamposPresupuesto.fecha] as String),
    color: Color(json[CamposPresupuesto.color] as int),
    recordatorio: (json[CamposPresupuesto.recordatorio] as int) == 1,
    idPersona: json[CamposPresupuesto.idPersona] as String,
    cantidad: json[CamposPresupuesto.cantidad] as double,
); // Presupuesto
```

Figura 6.9: Ejemplo de modelo de evento

En este apartado se definirá como se ha desarrollado el modelo de los objetos que persistirán en la base de datos.

El modelo utiliza un par de clases. Una donde se definen los diversos atributos que conforman, y se especifica si estos son necesarios o no para crear el objeto y donde se especifican los métodos para transformar el objeto a JSON y otra exclusivamente para definir las claves del JSON.

En la clase campos se define el nombre de los campos clave del JSON que utilizaremos en forma de variables finales, lo que nos permite separar el nombre del valor real de forma simple y eficaz a la hora de usar las tablas de la base de datos.

En la clase del objeto propiamente dicha es donde se utiliza el tipo de datos JSON para tanto insertar como leer de la base de datos, por lo que se han implementado métodos to y from JSON, como se puede observar en la figura 6.9.

Este trabajo nos permite simplificar el tratamiento con la base de datos, al tratar desde el modelo las conversiones y aportar el nombre de los campos. Esta simplificación se puede ver en la figura 6.10, donde se emplea constantemente la clase con los campos del JSON, CamposEvento, en este caso, para realizar la query. También podemos observar como se emplea el método fromJson para trasladar los datos desde la query al modelo.

Sobre la propia consulta, query, esta está realizada con argumentos, para evitar inyección, y a través de los métodos establecidos en la libreria SQLite.

En síntesis, este modelo permite una alta compenetración con la capa de base de datos gracias al tratamiento que se les da en el modelo y a que las consultas se realizan con los métodos aportados por la propia librería SQLite, lo que nos aporta seguridad y adaptabilidad.

```dart
Future<List<Evento>> leerEventoPorDia(DateTime day, DateTime dayFinal) async {
  final db = await instance.database;
  final orderBy = '${CamposEvento.fecha} ASC';

  final startOfDay = DateTime(day.year, day.month, day.day);
  final endOfDay = DateTime(dayFinal.year, dayFinal.month, dayFinal.day, 23, 59, 59);

  final maps = await db.query(
    tablaEventos,
    where: '${CamposEvento.fecha} >= ? AND ${CamposEvento.fecha} <= ? AND ${CamposEvento.idPersona} = ?',
    whereArgs: [startOfDay.toIso8601String(), endOfDay.toIso8601String(), _personaLogueada],
    orderBy: orderBy,
  );

  if (maps.isNotEmpty) {
    return maps.map((json) => Evento.fromJson(json)).toList();
  } else {
    throw Exception('Events not found for the specified day');
  }
}
```

Figura 6.10: Ejemplo de consulta a la BD

6.3.3. Base de datos

```dart
await db.execute('''
CREATE TABLE $tablaAjustes (
  ${CamposAjustes.id} $idType,
  ${CamposAjustes.idPersona} $textType,
  ${CamposAjustes.idioma} $textType,
  ${CamposAjustes.simboloMoneda} $textType,
  ${CamposAjustes.diaComienzoSemana} $integerType,
  FOREIGN KEY (${CamposAjustes.idPersona}) REFERENCES $tablaPersonas(${CamposPersona.email}),
  UNIQUE (${CamposAjustes.idPersona})
)
''');
```

Figura 6.11: Ejemplo de tabla de base de datos

En este apartado se comentará el código relativo a la base de datos y sus condiciones de ejecución.

Para la creación de la base de datos se comprueba al iniciar la aplicación si la propia base de datos existe en el path predeterminado de cada sistema, si existe se conecta a ella, en caso contrario, se ejecuta el script de creación, gracias al método de SQLite openDatabase, que, si no encuentra a la base de datos en la dirección, ejecuta el método que se le pasa en el atributo OnCreate.

El método OnCreate cuenta con instrucciones como la que se contempla en la figura 6.11, que crea la tabla ajustes de la base de datos con las restricciones correspondientes y nombres de los campos obtenidos del modelo.

Para activar las restricciones hace falta ejecutar instrucciones extra, por ejemplo, para las restricciones de clave foránea. Esto se hace mediante el atributo OnConfigure del propio método openDatabase, que llama a un método que ejecuta con la keyword PRAGMA la activación de las restricciones.

En definitiva, la base de datos se despliega y configura mediante tanto instrucciones para crear tablas tradicionales como con métodos incluidos en la librería SQLite.

6.3.4. Firebase

```
    await FirebaseAuth.instance.createUserWithEmailAndPassword(
  email: textoUsuario.text.trim(),
  password: textoContrasena.text.trim(),
);
```

Figura 6.12: Ejemplo de código de la librería firebase

A continuación se expondrá cómo se utiliza la librería firebase en el proyecto.

Como API REST con diversos servicios de almacenamiento, las llamadas a la librería firebase se realizan de una forma prácticamente idéntica a las que se realizan en la base de datos, con una función asíncrona que espera la respuesta al método invocado de la propia librería, en un patrón también singleton, como se ilustra en la figura 6.12.

Firebase cuenta, además, con su propio tipo de excepciones, lo cuál permite un tratamiento simplificado de los errores en caso de, por ejemplo, ya existir un usuario, ya que devuelve un tipo específico de error determinado según la función de conexión a la api utilizada.

El uso de firebase es, en síntesis, muy similar al realizado en la capa de modelo de nuestra aplicación, con funciones asíncronas que se conectan a la api mediante una instancia, es decir, el patrón sigleton.

CAPÍTULO 7
Producto desarrollado

En este capítulo se mostrará el producto finalmente desarrollado, esto es, las interfaces más relevantes finalmente elaboradas. También se explicará la funcionalidad que el usuario es capaz de desarrollar con estas interfaces, poniendo especial énfasis en las tareas descritas anteriormente en el apartado de escenarios.

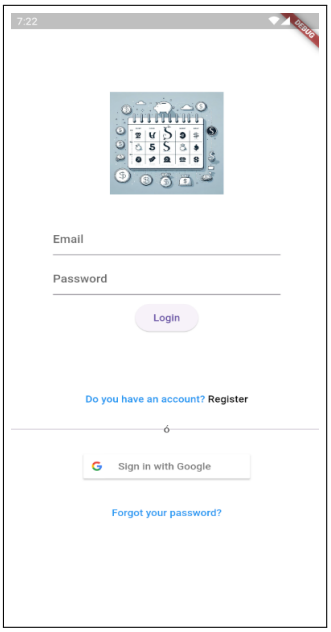


Figura 7.1: Formulario de inicio de sesión

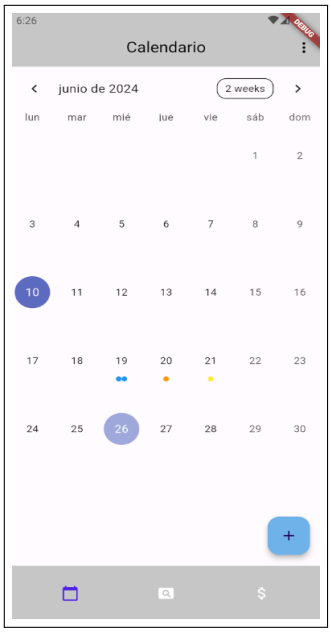


Figura 7.2: Calendario

El formulario de inicio de sesión mostrado en la figura 7.1 permite el login con un par de campos, email y contraseña. Además, gestiona, gracias al servicio de firebase, el mantener activa la sesión del usuario, redirigiéndolo a la pantalla principal, la de calendario, en caso de que esta no haya caducado.

Un par de enlaces permiten acceder tanto al formulario de registro como al de recuperación de contraseña. El formulario de registro es similar, pero con un campo adicional de repetición de la contraseña, para evitar errores. El formulario de recuperación de contraseña nos exige el correo electrónico para poder enviar el código de recuperación, y está gestionado por firebase.

Gracias también al servicio auth de firebase se ha implementado un botón de inicio de sesión con la cuenta de google, para facilitar a los usuarios el inicio de sesión y el registro a la aplicación.

El formulario de calendario mostrado en la figura 7.2 nos muestra la pantalla principal de la aplicación, con unos cuantos eventos planificados con un código de colores. Se pueden añadir más eventos en el día seleccionado con el botón flotante, que nos lleva al formulario de creación de eventos. También cuenta con la barra de navegación inferior, que permite navegar entre las listas de eventos y planificación. Al menú de ajustes se puede acceder con los tres puntos en la parte superior derecha de la pantalla. El calendario permite también marcar varios días para luego generar la lista de eventos con todos los días seleccionados al mismo tiempo en la pantalla con la lista de eventos.

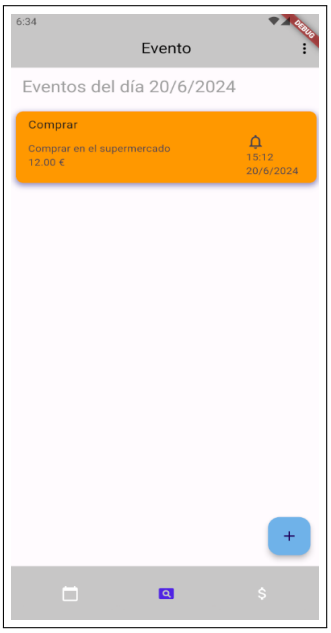


Figura 7.3: Lista de eventos

Figura 7.4: Formulario de creación de presupuestos

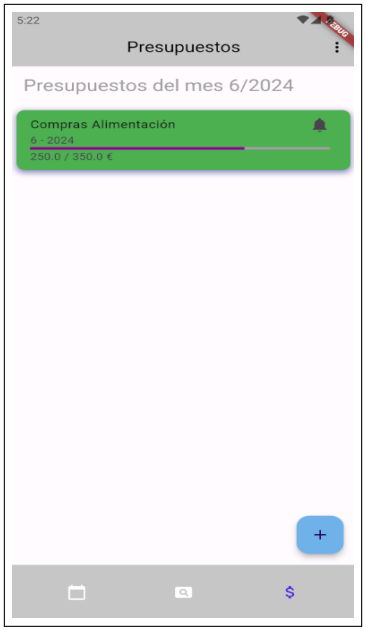


Figura 7.5: Lista de presupuestos

La lista de eventos, mostrada en la figura 7.3, nos muestra la lista de eventos. En la parte superior de la lista se puede observar en día en el que estamos, en caso de que no

hayamos seleccionado varios. El botón flotante nos llevará también, al igual que en la pantalla anterior, al formulario de creación de eventos.

En el propio evento podemos observar varias de sus características, como el gasto que supone, el título y la fecha, además de si las notificaciones están o no activadas gracias a la campanita.

Al pulsar en el evento se nos abren varias opciones, como editarlo, borrarlo o activar y desactivar las notificaciones, como se puede observar en la figura 7.6. El botón de edición nos lleva a la ventana de creación de evento, pero con los campos ya rellenados y con la acción del botón lanzando, en la capa de modelo, una instrucción de actualizar.

Estas opciones y formularios permiten las acciones descritas en el apartado de escenarios, siendo capaz con ellos tanto de observar el calendario para posterioremente ser capaz de preparar citas y activar notificaciones para ellas, tal y como se describe en el primer escenario, como de preparar presupuestos, como el que se puede observar en la figura 7.5, teniendo en cuenta los gastos que se han producido anteriormente, para poder organizar mejor las metas de ahorro en el futuro, como se describe en el escenario dos.

Figura 7.6: Lista de opciones de evento

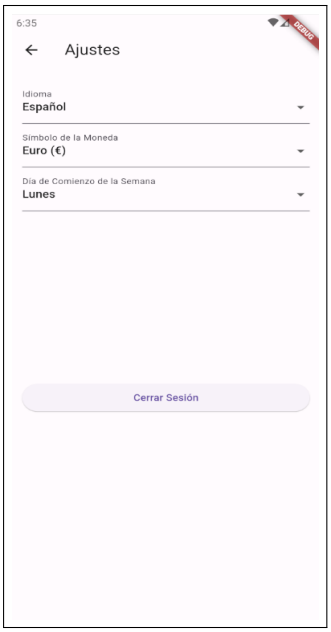


Figura 7.7: Formulario de los ajustes

En cuanto al formulario de ajustes, mostrado en la figura 7.7, permite gestionar varias opciones de personalización, como la moneda, el idioma o el día en que comienza la semana en el calendario. Además, permite cerrar sesión para alojar varias cuentas o usuarios.

Como conclusión, hemos recorrido una a una las pantallas, creadas a partir de los bocetos, más significativas de la aplicación, explicando la funcionalidad contenida en ellas que ha sido finalmente desarrollada.

CAPÍTULO 8
Validación

Para validar la aplicación se ha optado por probarla con usuarios reales. Estos usuarios seguirán una serie de tareas relacionadas con la aplicación, valorando si son capaces de realizarlas de forma ágil o si encuentran puntos de bloqueo, sustrayendo posibles problemas de diseño de estos puntos y proponiendo mejoras o soluciones.

8.1 Perfiles

A continuación se definirán los perfiles de las persona que han accedido a la aplicación en esta fase de evaluación de la misma, tratando uno en el que la persona que realiza la evaluación es similar a la persona arquetípica para la cuál se ha desarrollado la aplicación.

- El primer perfil es el de una mujer casada, de 48 años, con dos hijos y que trabaja en una escuela. Uno de sus hijos acude frecuentemente a actividades extraescolares relacionadas con el deporte.

  Durante su tiempo libre escucha música y el fin de semana lo dedica a realizar caminatas y senderismo por parajes naturales y urbanos.

  Le gustaría poder llegar de forma más olgada a fin de més, planificando sus gastos de forma eficiente.

  Este perfil es, de esta forma, similar al presentado en la definición de persona.

- El segundo perfil es el de un hombre soltero, de 41 años, que trabaja en un centro infantil.

  Durante su tiempo libre le gusta ver películas y realizar viajes relacionados con la cultura.

  Le gustaría poder ahorrar para realizar sus viajes y acudir más frecuentemente al cine.

8.2 Tareas

Las tareas a realizar serán tres, dos de ellas relacionadas con los escenarios propuestos y una útima relacionada con el apartado de ajustes de la aplicación.

- Esta tarea consta de preparar, como se propone en el escenario dos, un presupuesto, accediendo al formulario de creación de presupuestos, rellenándolo y confirmando su creación.

- En esta tarea se intenta preparar un evento y asignarlo al presupuesto previamente elaborado, de la misma forma, accediendo a su respectivo formulario, el de creación de eventos. Esta tarea está relacionada con el primer escenario.
- Por último se propone cambiar el día de comienzo de la semana en la ventana de ajustes de la aplicación.

8.3 Evaluación

Se procederá a continuación a explicar la experiencia con la aplicación de cada uno de los perfiles al realizar las tareas propuestas.

8.3.1. Primer perfil

El primer perfil no ha tenido problemas al registrarse en la aplicación, encontrando de forma rápida, aunque no directa, el formulario de registro.

A la hora de preparar el presupuesto, la primera tarea, le ha costado encontrar el formulario, buscándolo por varias otras pantallas antes de encontrar la de presupuestos y añadir el propio presupuesto.

A raíz de crear el presupuesto, ya no ha tenido problemas en realizar la segunda tarea, accediendo de forma directa a la pantalla de eventos y creando uno de forma rápida.

Por último, tampoco ha encontrado contratiempos al acceder a la pantalla de ajustes y cambiar el día, realizando así de forma satisfactoria la tercera tarea propuesta para la evaluación de la aplicación.

8.3.2. Segundo perfil

El segundo perfil ha optado por registrase a través de google, accediendo de forma instantánea a la aplicación.

A la hora de preparar el presupuesto, la primera tarea, le ha costado, al igual que al primer perfil, encontrar el formulario, intentando primero la creación de un evento, antes de rectificar y acceder al formulario correcto.

A raíz de crear el presupuesto, ya no ha tenido problemas en realizar la segunda tarea, al igual que el primer usuario, accediendo de forma directa a la pantalla de eventos, que ya había visitado previamente, y creando uno de forma rápida.

Por último, ha interactuado con las opciones del calendario brevemente antes de entrar al menú de ajustes para cambiar el día de comienzo de la semana, en el momento de realizar la tercera tarea.

8.4 Problemas encontrados y mejoras propuestas

Las conclusiones sustraídas de la evaluación son que los usuarios son capaces de realizar de forma simple y eficaz las tareas una vez conocen la distribución tanto de dónde se encuentran los presupuesto como los eventos en la aplicación.

Para solventar los problemas iniciales que estos encuentran a la hora de realizar la primera tarea, se propone hacer un tutorial, recorrido inicial, la primera vez que se inicializa la aplicación.

Otra propuesta sería el señalizar de forma más clara en que parte de la aplicación se encuentra el usuario, si en la relacionada con eventos o en la relacionada con presupuestos.

CAPÍTULO 9
Conclusiones

En este trabajo se ha llevado a cabo el desarrollo desde cero de una aplicación móvil que permite afrontar, a través de una gestión eficiente tanto del tiempo como de los recursos económicos, el mundo de hoy día. Se ha logrado la creación de una herramienta eficaz para gestionar, de forma holística, tanto el tiempo como el dinero.

El enfoque, centrado en el usuario, ha permitido otorgar a estos una plataforma intuitiva y fácil de usar para su organización personal, brindándoles una solución integrada a sus problemas a la hora de afrontar la gestión de sus recursos.

9.1 Objetivos

Al inicio del desarrollo de la aplicación, se establecieron tres objetivos principales. Ahora evaluaremos si se han cumplido plenamente, parcialmente o si no se han alcanzado de manera satisfactoria.

- Desarrollar un Calendario Personalizado: Este objetivo ha sido alcanzado, ya que el usuario es capaz de personalizar funciones dentro del calendario y de visualizar fácilmente los eventos que están por venir, ayudandolo a visualizar como distribuye su tiempo.

- Implementar un Rastreador de Gastos: Este objetivo ha sido cumplido con la lista personalizable a la que añadir gastos, permitiendo además modificar los días que este desea incluir a la lista. Además puede categorizarlos usando diferentes métodos, desde eligiendo nombre o color a asignándolos a un presupuesto.

- Crear un Planificador Financiero: Este objetivo se ha abordado parcialmente, permitiendo la creación de presupuestos que permiten organizar los gastos y fijarse metas financieras, pero sin añadir mayor profundidad a la planificación.

En conclusión, los objetivos se han alcanzado de forma satisfactoria a la hora de desarrollar la aplicación, habiendo completado dos de los tres objetivos de forma completa y uno de ellos de forma parcial.

9.2 Trabajo futuro

El trabajo futuro principal sería añadir las mejoras a la interfaz que se proponen en el apartado de validación, en el siguiente ciclo de desarrollo DCU.

Además, se podría implementar más profundidad a la planificación, introduciendo estádísticas y modelos predictivos según el més, otorgando a la aplicación una característica potente a la hora de gestionar economicamente los recursos de sus usuarios.

Como añadido, flutter permite la creación de forma sencilla para múltiples plataformas, por lo que es posible preparar la aplicación para desplegarla en otros sistemas operativos o en la web de forma relativamente sencilla.

Para finalizar, ya se usa el servicio de autentificación de firebase en la aplicación, pero se podría adaptar para usar el servicio de base de datos, permitiendo la persistencia de datos entre diferentes dispositivos de forma simple y en tiempo real, coordinandolos de forma directa.

Bibliografía

[1] Donald A. Norman. *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books, 2013.

[2] Steve Krug. *Don't Make Me Think: A Common Sense Approach to Web Usability*. New Riders Publishing, 2000.

[3] Alan Cooper. *The Inmates are Running the Asylum*. Sams, 1999.

[4] Alan Dennis, Barbara Haley Wixom, and David Tegarden. *Systems Analysis and Design with UML*. John Wiley Sons, 2015.

[5] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems*. Pearson, 2016.

[6] Victor Ahmad. Build a flutter app using mvvm, 2023. Disponible en: `https://medium.com/@Victor.Ahmad/implementing-model-view-viewmodel-mvvm-in-a-flutter-app-a-step-by-step-guide-92b05e6e8192`, Consultada el 27 de junio de 2024.

[7] Leonard Richardson, Mike Amundsen, and Sam Ruby. *RESTful Web APIs: Services for a Changing World*. O'Reilly Media, 2013.

[8] Tim Bray. *The JSON Saga*. O'Reilly Media, 2017.

[9] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.

[10] Mike Owen. *The Definitive Guide to SQLite*. Apress, 2010.

[11] Google Developers. Firebase documentation, 2023. Disponible en: `https://firebase.google.com/docs`.

[12] Kathy Braun and Shailen Rahman. *Dart: Up and Running*. O'Reilly Media, 2014.

[13] Jeff Kasper and et al. *Beginning App Development with Flutter: Create Cross-Platform Mobile Apps*. Apress, 2020.

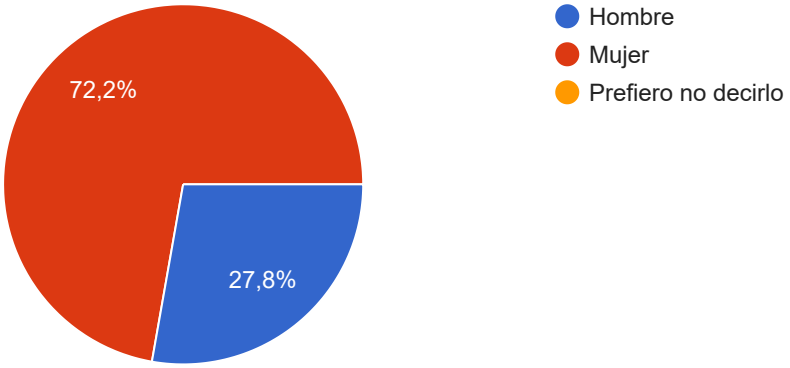
Anexo A

Encuesta de uso de aplicaciones

Preguntas Demográficas

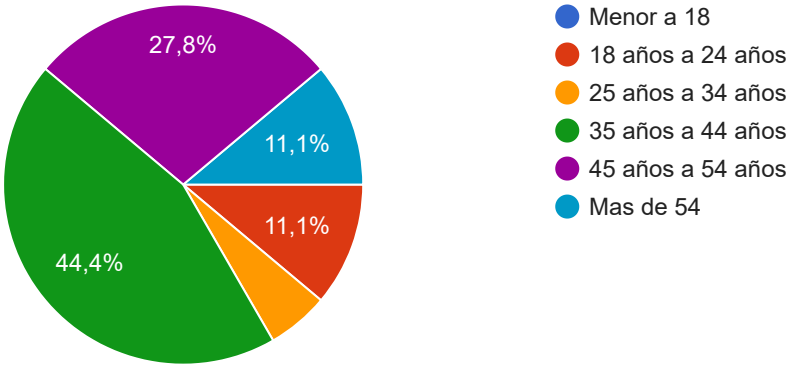
Sexo

18 respuestas



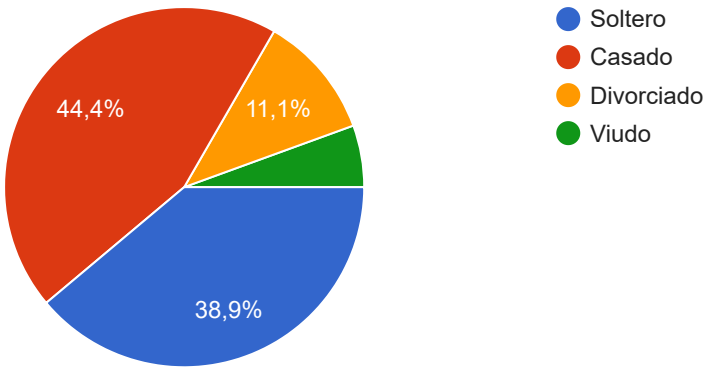
Edad

18 respuestas



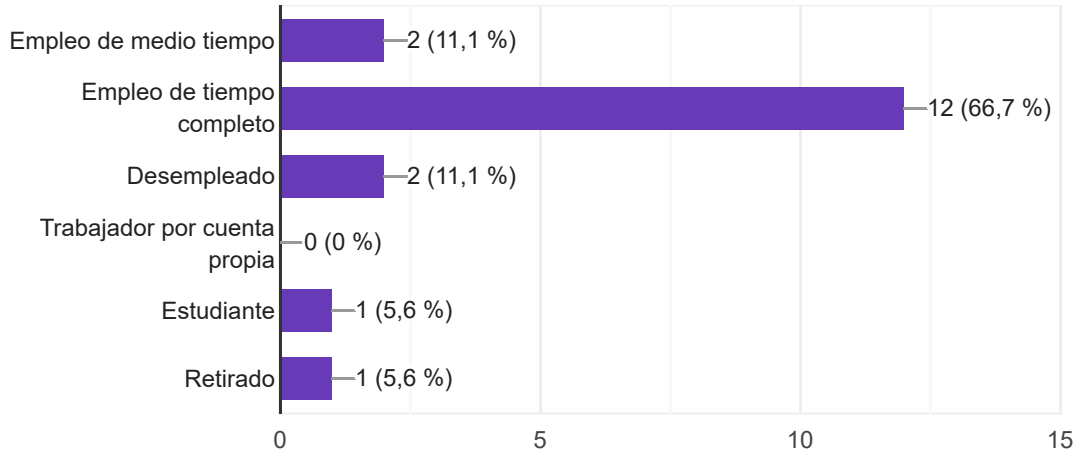
¿Cuál es tu estado civil?

18 respuestas



¿Cuál es tu situación laboral actual?

18 respuestas



¿Cuál es tu trabajo? ¿Qué estudias?

16 respuestas

Desarrollador de software

Turismo

Cocinera

artesana

Asesora de seguros de asosaciones

Profesor Música

Fisioterapeuta

Ama de casa

Funcionaria

Guardia Civil

Funcionario

Professora de secundaria

Maestra

Marketing

Docente

Funcionaria.Soy Secretaria de ayuntamiento

¿Tienes algún pasatiempo o afición?

16 respuestas

Si

Cocinar, videojugar, jueguitos de mesa y leer

Jugar a videojuegos. Jugar y ver fútbol. Pasar tiempo con mi familia, amigos y mascotas. Hacer senderismo o running.

Costura

pintar

Caminar

Sí

Lectura, música, deporte

Caminar

La agricultura

Bailar, leer

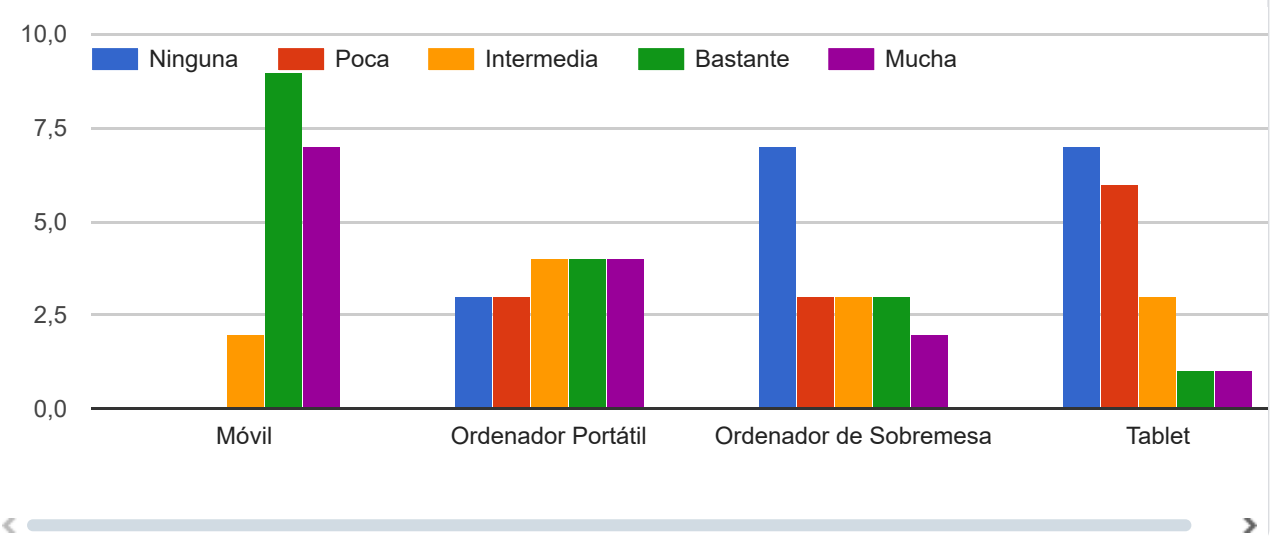
Cine

Cine,música,teatro

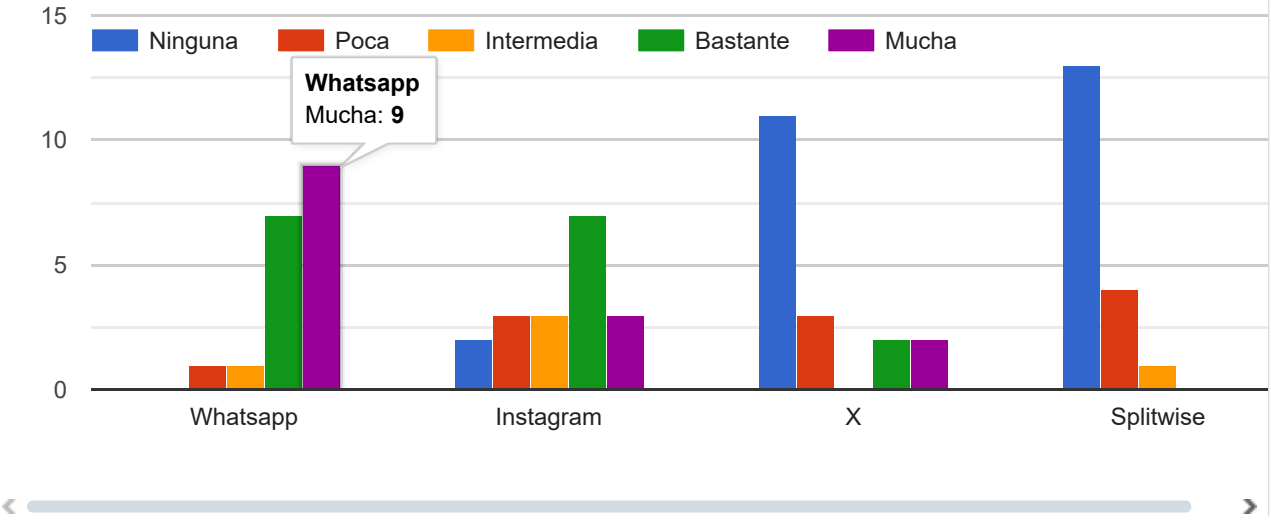
Música, deporte, lectura

Preguntas Tecnológicas

¿Con qué frecuencia utilizas los siguientes dispositivos?



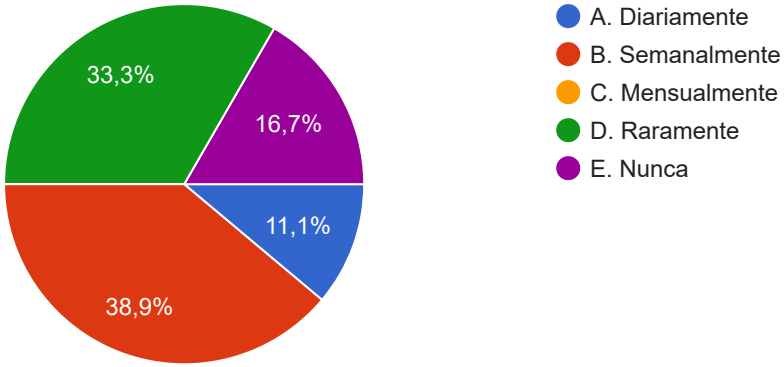
¿Has usado alguna de estas aplicaciones?¿Con qué frecuencia?



Preguntas Específicas

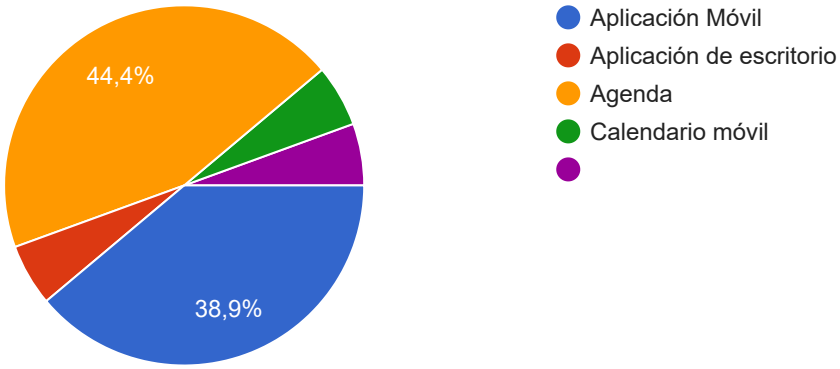
¿Con qué frecuencia utilizas aplicaciones o herramientas para gestionar tu tiempo y planificar tus actividades diarias?

18 respuestas



¿Qué sueles usar para gestionar tu tiempo y planificar tus actividades diarias?

18 respuestas



¿Tienes algún inconveniente y/o problema en tus métodos actuales para gestionar tu tiempo y planificar tus actividades diarias con tu método actual?

17 respuestas

No

Que básicamente recurro a la memoria y a veces se me olvidan cosas, quizás unas notificaciones en el movil podrían solucionar eso.

Sí, me cuesta mucho ponerme a hacer el TFG aunque en la agenda ponga que tengo que hacerlo.

Se podría mejorar

no

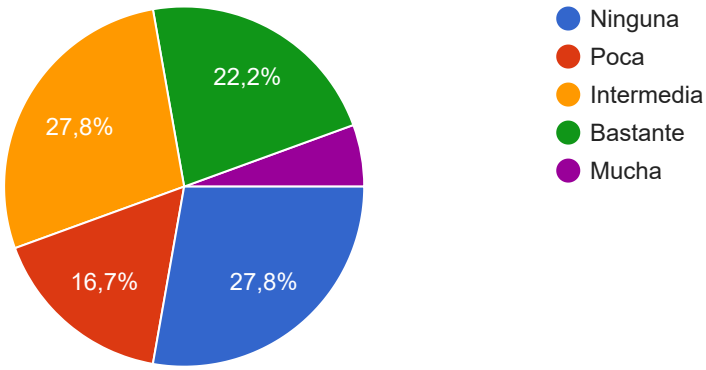
No

Es muy simple

Si, me falta interconexion entre aplicaciones y dispositivos. Tengo una agenda para escribir en la tablet (en good notes) y uso tambien el calendar de movil, y no puedo sincronizar

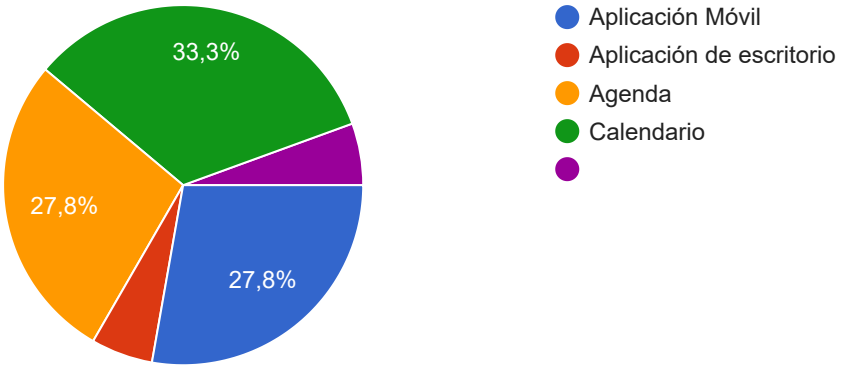
¿Con qué frecuencia utilizas aplicaciones u otros métodos para planificar tu calendario?

18 respuestas



¿Qué sueles usar para planificar tu calendario?

18 respuestas



¿Tienes algún inconveniente y/o problema en tus métodos actuales para planificar tu calendario con tu método actual?

16 respuestas

No

No muchos

no

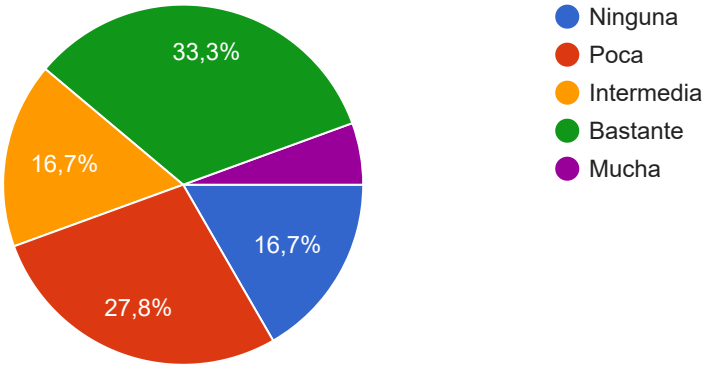
No

Se queda corto

me falta interconexion entre aplicaciones y dispositivos. Tengo una agenda para escribir en la tablet (en good notes) y uso tambien el calendar de movil, y no puedo sincronizar

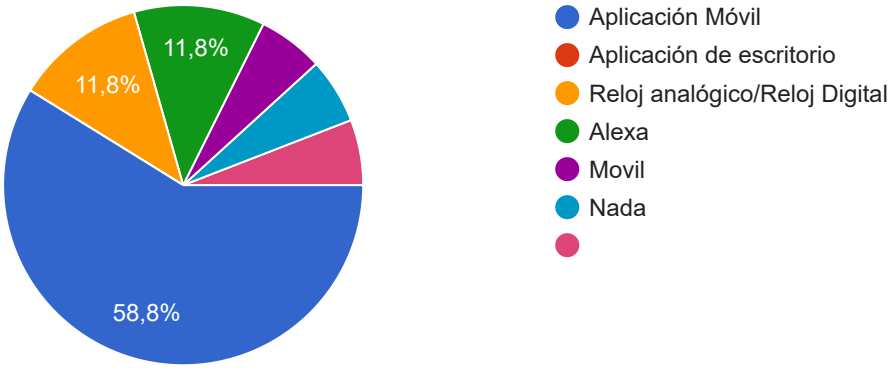
¿Con qué frecuencia utilizas aplicaciones o relojes para establecer recordatorios o alarmas?

18 respuestas



¿Qué sueles usar para establecer recordatorios o alarmas?

17 respuestas



¿Tienes algún inconveniente y/o problema en tus métodos actuales para establecer recordatorios o alarmas con tu método actual?

15 respuestas

No

La verdad que no

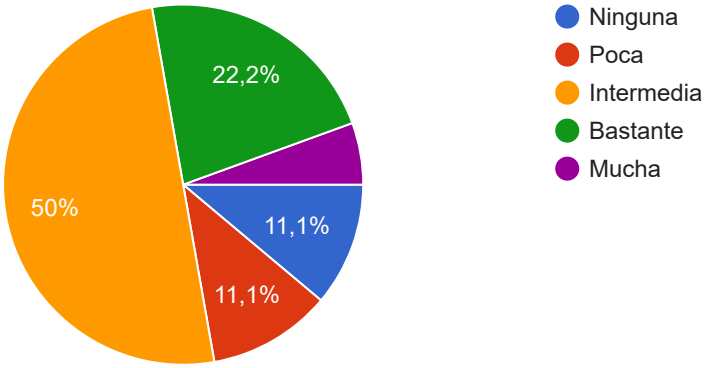
no

No

Es muy simple

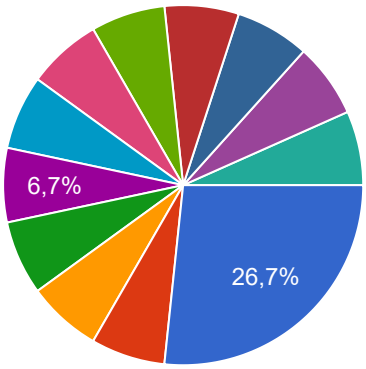
¿Cuánta atención prestas al tiempo que empleas en realizar tareas?

18 respuestas



¿Cronometras de alguna forma el tiempo que tardas?

15 respuestas



- Aplicación Móvil
- Aplicación de escritorio
- Cronómetro analógico
- Miro el reloj y lo vuelvo a mira…
- Mirando la hora
- No
- ninguno reloj
- Reloj

1/2

¿Tienes algún inconveniente y/o problema a la hora de medir el tiempo que tardas con tu método actual?

13 respuestas

No

A veces se me olvida mirar antes de empezar xd

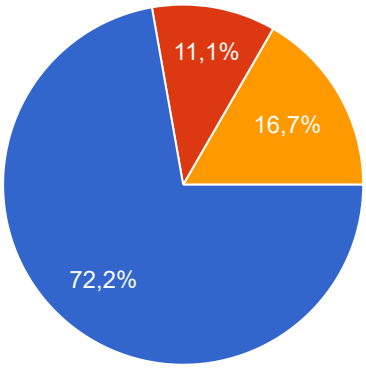
no

No

No uso metodos digitales para controlarlo

¿Con qué frecuencia usas aplicaciones o otros métodos para establecerte metas y objetivos ?

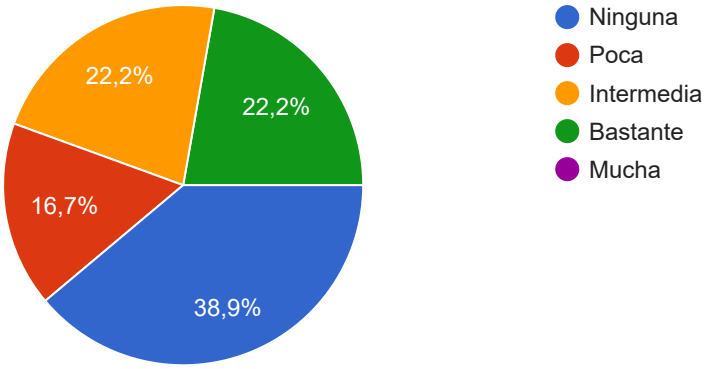
18 respuestas



- Ninguna
- Poca
- Intermedia
- Bastante
- Mucha

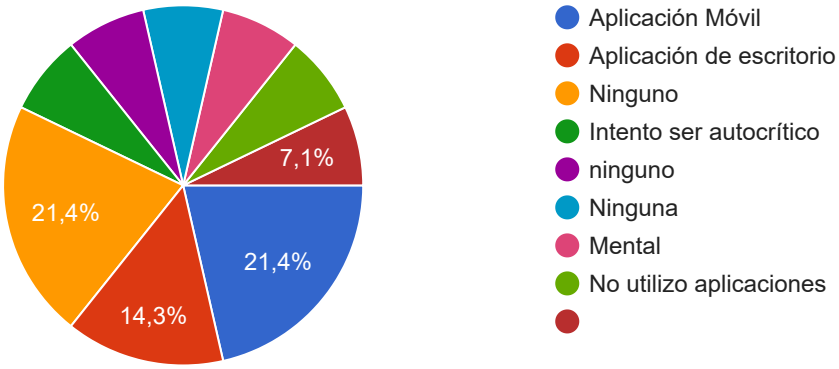
¿Con qué frecuencia prestas atención a tu productividad y rendimiento?

18 respuestas



¿Qué método usas para medir tu productividad y rendimiento?

14 respuestas



¿Tienes algún inconveniente y/o problema para medir tu productividad y rendimiento con tu método actual?

13 respuestas

No

Igual unos gráficos estarían chulos.

No realmente, intento no ser duro conmigo mismo

No

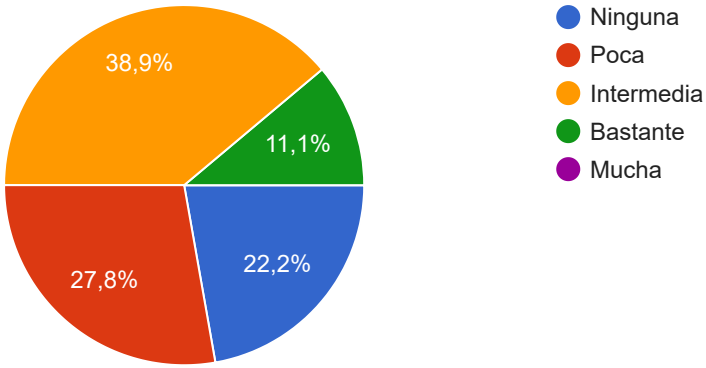
a veces

No

NO

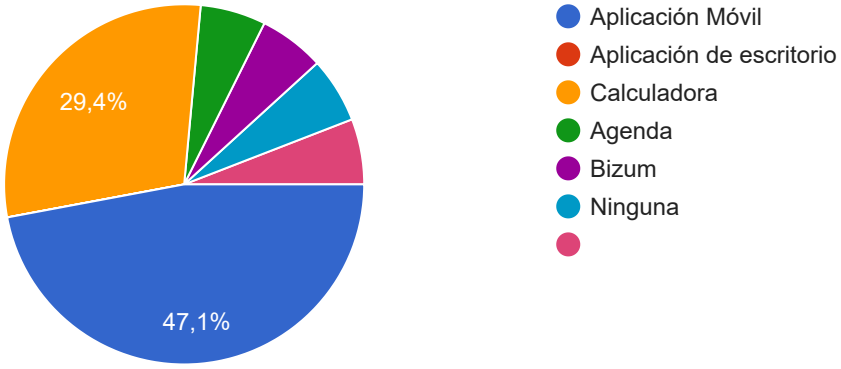
¿Con qué frecuencia repartes pagos con personas de tu entorno?

18 respuestas



¿Qué método usas repartir los pagos?

17 respuestas



¿Tienes algún inconveniente y/o problema para repartir los pagos con tu método actual?

15 respuestas

No

No

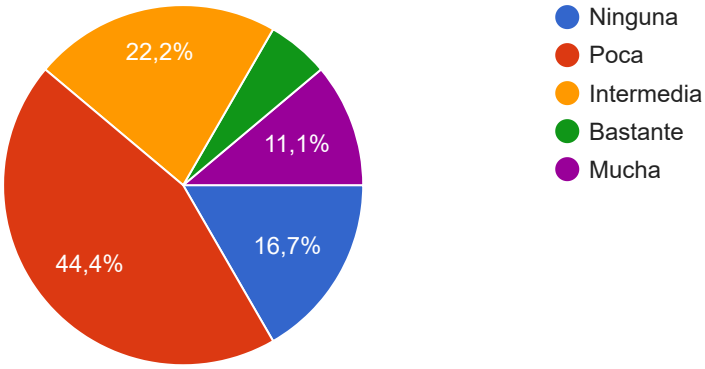
No se realizan los pagos automáticos

Limite de números de pagos por la aplicación

NO

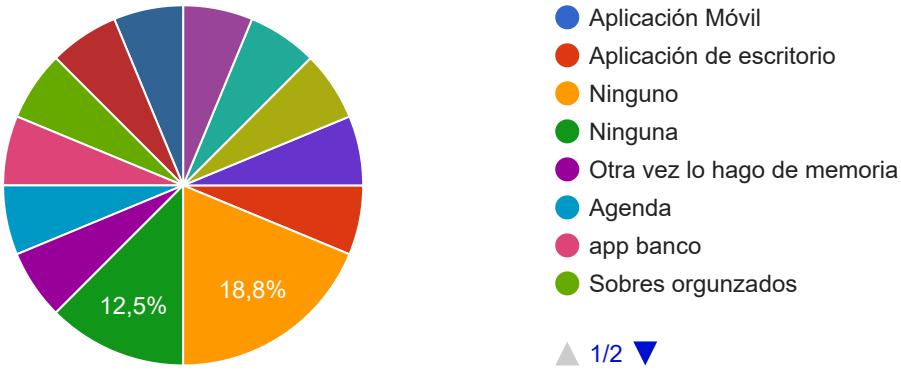
¿Con qué frecuencia planificas tus gastos mensuales?

18 respuestas



¿Qué método usas para planificar tus gastos

16 respuestas



¿Tienes algún inconveniente y/o problema para planificar tus gastos con tu método actual?

13 respuestas

No

Por el momento no, pero acabará habiendo

no

No

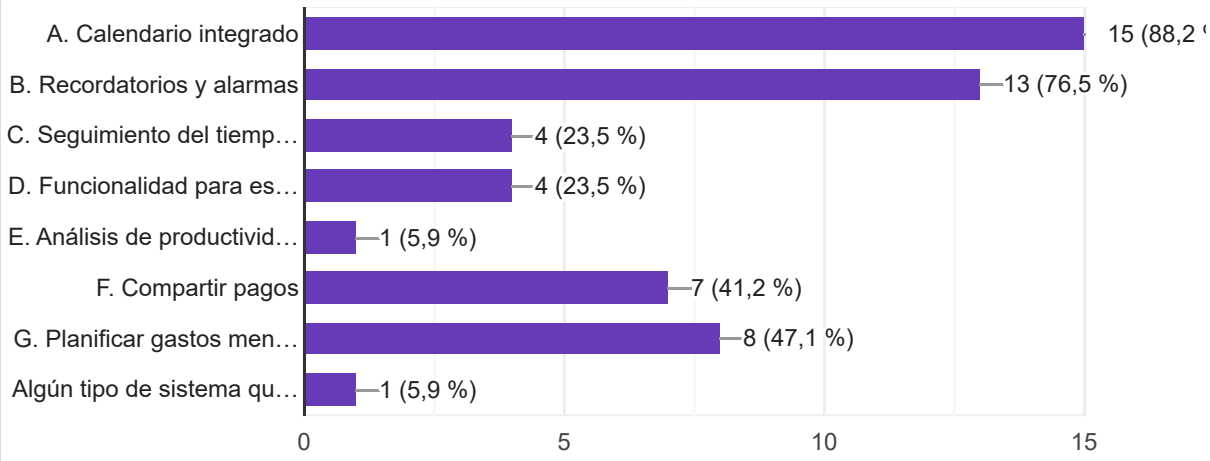
Poco exhaustivo

Ni

NO

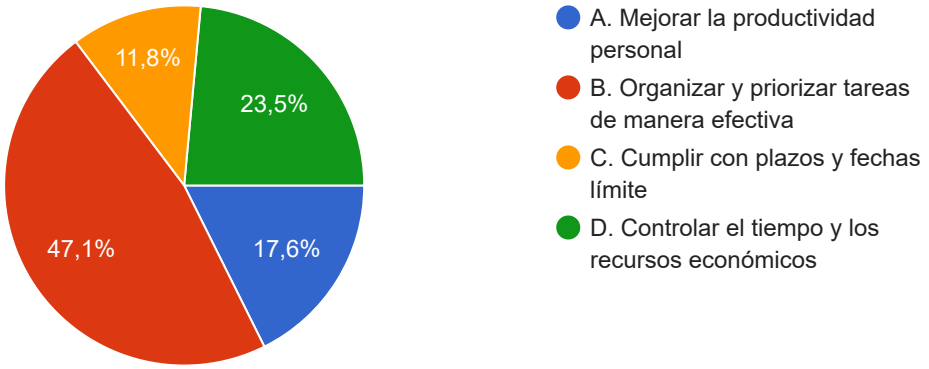
¿Qué características consideras más importantes en una aplicación de gestión de tiempo y planificación económica? (Selecciona todas las que apliquen)

17 respuestas



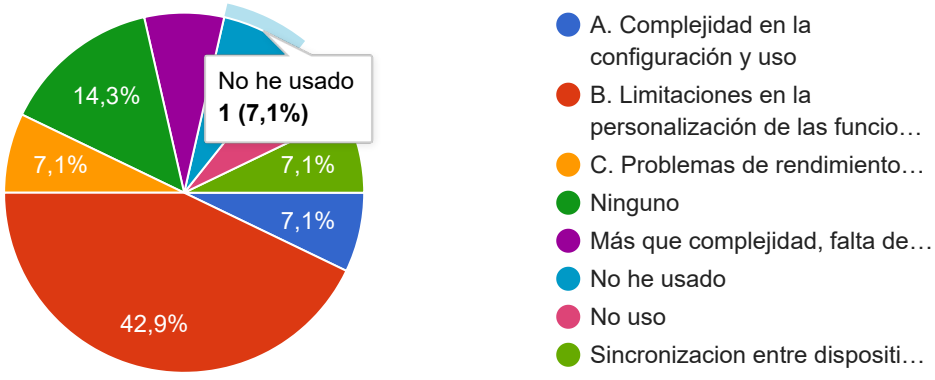
¿Qué te motivaría a utilizar una aplicación de gestión de tiempo y planificación económica?

17 respuestas



¿Qué desafíos o problemas has experimentado con aplicaciones de gestión de tiempo y planificación en el pasado?

14 respuestas



¿Tienes alguna característica específica en mente que crees que sería útil en una aplicación de gestión de tiempo y planificación, pero que no has encontrado en otras aplicaciones?

11 respuestas

No

Notificaciones recurrentes conforme se acerca la deadline, no sé si hay aplicaciones que lo hacen pero las que yo lo he usado te mandan una, que quitas porque en ese momento no te apetece que esté, no te mandan ninguna más y se te olvida.

Se podría crear un apartado a modo de foro en el que la gente comparta sus modelos de trabajo o gestión de tiempo, para que aquellas personas que no estén muy familiarizadas con el tema o quieran probar otras metodologías.

Planificación menú de comidas por semanas .

No

¿Cómo te gustaría que una aplicación de gestión de tiempo y planificación te ayudara en la planificación de tus recursos económicos?

9 respuestas

No se

La especifique anteriormente, el sistema de añadido de gastos por categorías. Por poner ejemplos: gasto estático (alquiler, facturas de gastos), gastó en comida, gastó en ocio, gastó en caprichos, inversiones...

Control gastos por semanas.

No sé

Con avisos al pasar los presupuestos establecidos

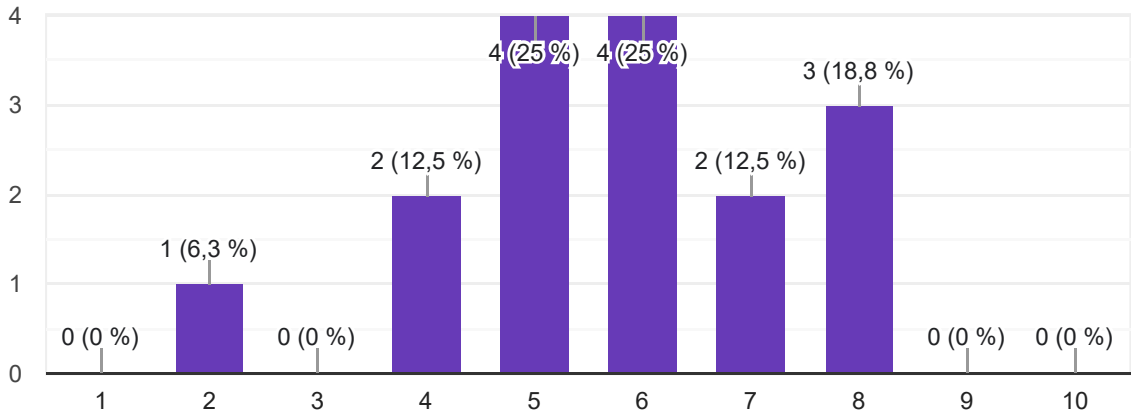
Lo más sencilla posible

Sinceonizandose con los datos les banco

No lo se

En una escala del 1 al 10, ¿cómo calificarías tu satisfacción general con las aplicaciones de gestión de tiempo y planificación que has utilizado hasta ahora? (Siendo 1 muy insatisfecho y 10 muy satisfecho)

16 respuestas



ANEXO

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
• **Fin de la pobreza.**		X		
• **Hambre cero.**			X	
• **Salud y bienestar.**		X		
• **Educación de calidad.**			X	
• **Igualdad de género.**				X
• **Agua limpia y saneamiento.**				X
• **Energía asequible y no contaminante.**				X
• **Trabajo decente y crecimiento económico.**		X		
• **Industria, innovación e infraestructuras.**				X
• **Reducción de las desigualdades**.			X	
• **Ciudades y comunidades sostenibles**.				X
• **Producción y consumo responsables.**			X	
• **Acción por el clima.**				X
• **Vida submarina.**				X
• **Vida de ecosistemas terrestres.**				X
• **Paz, justicia e instituciones sólidas.**				X
• **Alianzas para lograr objetivos.**				X

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

Este trabajo de fin de grado aborda la gestión económica a nivel familiar y se alinea con varios Objetivos de Desarrollo Sostenible (ODS) que son fundamentales para el progreso y bienestar de la sociedad. La optimización de la economía familiar, la mejora de la salud y bienestar, y el fomento del trabajo decente y el crecimiento económico son aspectos esenciales que esta investigación y su aplicación práctica buscan mejorar de manera significativa.

El primero de los ODS es el fin de la pobreza. Este TFG, al centrarse en la gestión económica de las familias, facilita la optimización de sus recursos financieros. Mediante estrategias y herramientas que promueven el ahorro y una mejor administración del dinero, las familias pueden reducir el riesgo de caer en la pobreza. La aplicación desarrollada permite a las familias monitorizar y controlar sus gastos, identificar áreas donde pueden ahorrar y establecer metas financieras realistas. Al hacerlo, se crea una base sólida para mejorar su estabilidad económica a largo plazo, contribuyendo directamente al objetivo de erradicar la pobreza en todas sus formas.

El segundo ODS relacionado es el de salud y bienestar. Una buena planificación económica también tiene un impacto positivo en la salud mental y física de las personas. La incertidumbre financiera es una de las principales fuentes de estrés, lo cual puede derivar en problemas de salud serios. Esta aplicación no solo ayuda a las familias a gestionar su dinero, sino también a organizar su tiempo. La planificación temporal que ofrece permite a las personas distribuir su tiempo de manera equilibrada entre trabajo, responsabilidades y tiempo libre. Esto crea oportunidades para el descanso y la relajación, esenciales para prevenir el estrés y mejorar el bienestar general. Al reducir la ansiedad relacionada con la gestión del tiempo y el dinero, las familias pueden disfrutar de una mejor calidad de vida.

El tercer ODS es el de trabajo decente y crecimiento económico. Al fomentar el ahorro, esta aplicación permite a las familias acumular capital que pueden invertir en diversas áreas. Estas inversiones pueden ir desde la educación y el desarrollo profesional hasta la creación de pequeños negocios o la compra de bienes duraderos. La capacidad de invertir y generar ingresos adicionales es crucial para el crecimiento económico sostenido. Además, cuando las familias tienen un mejor control sobre sus finanzas, pueden tomar decisiones más informadas y responsables que no solo beneficien su economía individual, sino también la economía en general. Un mayor nivel de ahorro y una mejor gestión de los recursos económicos a nivel familiar pueden llevar a una mayor estabilidad y prosperidad económica a nivel comunitario y nacional.

En resumen, este TFG tiene un impacto multifacético que va más allá de la simple gestión económica. Al mejorar la economía familiar, se contribuye a la erradicación de la pobreza, se promueve el bienestar y la salud de las personas y se fomenta un crecimiento económico sostenible y trabajo decente. Estos efectos combinados muestran cómo las herramientas y estrategias de gestión económica pueden ser poderosos agentes de cambio en la sociedad, ayudando a alcanzar los objetivos globales establecidos por la ONU para un desarrollo sostenible.