

WalletMoneyFamily: Aplicación web para control de gastos familiares

Tabla de contenido

Contenido	Pág.
Introducción.....	3
Contextualización de la necesidad	4
Planteamiento del problema	4
Fase de planificación del ciclo de vida del desarrollo de software (Fase de Análisis de Proyectos)	5
Alcance	5
Estructura del Desglose del Trabajo (EDT)	6
Planeación.....	6
Análisis	7
Diseño.....	7
Desarrollo	7
Pruebas.....	8
Despliegue	8
Objetivos.....	8
General	8
Específicos	9
Metodología ágil seleccionada	9
Justificación del Alcance	9
Repositorio.....	12
Fase de Análisis del ciclo de vida del desarrollo de software (Fase de Planeación de Proyectos)	12
Levantamiento de Información (Herramientas y Conclusiones)	12
Historias de usuario	14
Requerimientos funcionales (RQF) y no funcionales (RQNF)	15
Fase de Diseño del ciclo de vida del desarrollo de software (Fase de Ejecución de Proyectos)	16
Prototipos de interfaz.....	16
Prototipo de Baja Fidelidad	16
Prototipo de Alta Fidelidad	16
Modelos de comportamiento y estructura	19
Pruebas Usabilidad	27
Fase de Desarrollo del ciclo de vida del desarrollo de software (Fase de Ejecución de Proyectos)	30

Prototipo Funcional	31
Enlace de Repositorio GitHub	33
Conclusión	38
Referencias Bibliográficas	39

Tabla de Tablas

Tablas	Pág.
Tabla 1 Matriz de Riesgos	11
Tabla 2 Presupuesto para el cliente.....	11
Tabla 3 Conclusiones.....	12
Tabla 4 Historias de Usuario.....	14
Tabla 5 Requerimientos funcionales (RQF)	15
Tabla 6 Requerimientos no funcionales (RQNF)	15
Tabla 7 Casos de Uso Documentados.....	19
Tabla 8 Tareas Realizadas.....	27
Tabla 9 Resultados Cuantitativos.....	28
Tabla 12 Patrones Arquitectónicos y de Diseño Aplicados	31
Tabla 13 Avances de Codificación	33

Tabla de Imágenes

Figuras	Pág.
Figura 1 Mapa de Stakeholders (Influencia vs Interés)	10
Figura 2 Diagrama de Flujo de Solución Seleccionada	14
Figura 3 User Map	17
Figura 4 User Flow	18
Figura 5 Diagrama Casos de Uso	21
Figura 6 Diagrama de Secuencia	22
Figura 7 Diagrama de clases	23
Figura 8 Diagrama de Componentes	24
Figura 9 Diagrama de Despliegue	25
Figura 10 Diagrama Relacional	26
Figura 11 Comentarios cualitativos (usuarios reales)	29
Figura 12 Diagrama de Arquitectura	30
Figura 13 Estructura del directorio del proyecto	32
Figura 14 Formulario pregunta 1	34
Figura 15 Formulario pregunta 2	34
Figura 16 Formulario pregunta 3	35
Figura 17 Formulario pregunta 4	35
Figura 18 Formulario pregunta 5	36
Figura 19 Formulario pregunta 6	36
Figura 20 Formulario pregunta 7	37
Figura 21 Formulario pregunta 8	37
Figura 22 Formulario pregunta 9	37

Introducción

El proyecto WalletMoneyFamily responde a una necesidad crítica identificada en la familia Aldana Gómez: la falta de un sistema estructurado y centralizado para la gestión de las finanzas del hogar. Actualmente, la familia utiliza métodos informales y desorganizados, como anotaciones dispersas en cuadernos, hojas de cálculo sin formato definido o, en muchos casos, la ausencia total de registros. Esta situación genera dificultades para rastrear los ingresos y gastos, lo que frecuentemente resulta en desequilibrios financieros, donde los gastos superan los ingresos disponibles, limitando la planificación económica y causando tensiones en la toma de decisiones familiares. Este problema es común en muchos hogares que carecen de herramientas centralizadas para la gestión financiera. Para abordar esta necesidad, WalletMoneyFamily es una aplicación web diseñada para permitir a las familias registrar, monitorear y analizar sus ingresos y gastos de manera sistemática. Desarrollada por un desarrollador junior en un plazo de 6 a 8 semanas, con un costo para el cliente de aproximadamente 1859 USD (113 horas a 10-20 USD/hora, con 10% de contingencia), basado en tarifas junior y un presupuesto del desarrollador cercano de 25 USD utilizando herramientas gratuitas y de bajo costo, la aplicación emplea Flask para el backend, SQLite o MongoDB para el almacenamiento de datos, Bootstrap para una interfaz responsiva y Chart.js para reportes visuales. Gestionado mediante un tablero Kanban en Trello con las columnas To Do, In Progress y Done, el proyecto asegura un desarrollo ágil alineado con las fases del Ciclo de Vida del Desarrollo de Software (SDLC). La aplicación consta de dos módulos principales: autenticación (registro e inicio de sesión con Flask-Login) y registro/seguimiento de ingresos y gastos (visualización de datos financieros por miembro de la familia). Al ofrecer una plataforma segura, accesible y fácil de usar, WalletMoneyFamily busca prevenir desequilibrios financieros, reducir conflictos y mejorar la toma de decisiones informadas, contribuyendo a la estabilidad financiera y la calidad de vida de la familia Aldana Gómez.

Contextualización de la necesidad

En la familia Aldana Gómez se evidencia una falta de registro y control estructurado de sus gastos. Actualmente, el manejo de las finanzas se realiza de forma informal y desorganizada, ya sea mediante anotaciones aisladas en cuadernos, registros en hojas de cálculo sin un formato definido o, en muchos casos, sin ningún tipo de registro. Esta situación genera dificultades para llevar un seguimiento claro de los ingresos y egresos del hogar, lo que con frecuencia provoca que los gastos superen los ingresos disponibles, generando desequilibrios financieros, limitaciones en la planeación económica, y tensiones en la toma de decisiones dentro del núcleo familiar. Este problema es común en muchas familias, donde la falta de un sistema centralizado para monitorear ingresos y gastos lleva a decisiones poco informadas. De esta necesidad surge WalletMoneyFamily, una aplicación web diseñada para que las familias puedan registrar, monitorear, y analizar sus ingresos y gastos de forma estructurada. WalletMoneyFamily permitirá a los usuarios registrar ingresos y gastos asociados a cada miembro y visualizar tendencias financieras mediante tablas y gráficos, todo desde una interfaz web accesible y segura, contribuyendo al equilibrio financiero del hogar y mejorando la calidad de vida del todo el núcleo familiar.

Planteamiento del problema

¿Cómo una aplicación web puede organizar y centralizar el registro y análisis de los ingresos y gastos familiares, permitiendo identificar quién genera cada ingreso y quién realiza cada gasto, con el propósito de prevenir desequilibrios financieros, reducir conflictos y mejorar la toma de decisiones, fomentando así un equilibrio económico en el hogar?

Fase de planificación del ciclo de vida del desarrollo de software (Fase de Análisis de Proyectos)

Alcance

El proyecto WalletMoneyFamily consiste en desarrollar una aplicación web sencilla, segura y accesible desde cualquier navegador para que la familia Aldana Gómez organice, centralice y analice sus ingresos y gastos, identificando quién genera cada ingreso y quién realiza cada gasto, con el fin de evitar desequilibrios financieros, conflictos familiares y decisiones poco informadas, promoviendo un equilibrio financiero en el hogar. La aplicación será desarrollada por un desarrollador junior individual en 6-8 semanas, con un presupuesto real para el desarrollador de aproximadamente 25 USD (usando herramientas gratuitas y de bajo costo) y un costo para el cliente de aproximadamente 1859 USD (113 horas a 10-20 USD/hora, con 10% de contingencia), basado en tarifas junior. Se utilizará la metodología ágil Kanban, gestionada en un tablero Trello con columnas To Do, In Progress y Done, actualizado diariamente para el informe del proyecto. El entorno de desarrollo será Visual Studio Code con extensiones como Python, Pylance, GitLens y Live Server para codificar, depurar y gestionar Flask, HTML/CSS y JavaScript. Las tecnologías incluyen Flask para el backend, SQLite o MongoDB para la base de datos (tablas: usuarios, ingresos, gastos), Bootstrap para una interfaz responsiva, y Chart.js para gráficos de barras y pastel. El control de versiones se gestionará en un repositorio GitHub con rama main y ramas por funcionalidad, y la aplicación se desplegará en Heroku o Render con integración CI/CD. El alcance abarca 2 módulos principales; el módulo 1 es para la autenticación (registro e inicio de sesión con Flask-Login con usuario/correo y contraseña), y el módulo 2 es para registro de ingresos (monto, categoría, título, descripción, fecha) y gastos (monto, categoría, título, descripción, fecha), y visualización mediante tablas y gráficos de tendencias financieras. Los entregables incluyen la aplicación desplegada, formularios funcionales, reportes visuales, un

repositorio GitHub con README.md, un tablero Trello aplicando la metodología ágil Kanban, un informe de proyecto de software en formato APA (PDF) que detalle contextualización, planteamiento del problema, objetivos (general: mejorar la gestión financiera; específicos: implementar autenticación, registro, reportes, despliegue, y documentación), alcance, metodología, matriz de riesgos, mapa de stakeholders (gráfico de dispersión), requerimientos funcionales (RQF1: autenticación, RQF2: registro, RQF3: reportes), requerimientos no funcionales (RQNF1: respuesta <2s, RQNF2: interfaz responsiva, RQNF3: uptime >99%, RQNF4: contraseñas con hash), historias de usuario (HU1-HU3), diagrama de flujo (Draw.io), arquitectura (UML), prototipos de interfaz (Figma/papel), resultados de pruebas, y evidencias de despliegue. Las pruebas incluyen unitarias (pytest para lógica de autenticación y registro) y funcionales (manuales para validar autenticación, registro y visualización). El proyecto no incluye integraciones con bancos, notificaciones en tiempo real, pruebas de carga extensivas ni aplicaciones móviles, debido a limitaciones de tiempo, presupuestales y técnicas. Las restricciones son un presupuesto real para el desarrollador junior de aproximadamente 25 USD, un desarrollador junior con 10-15 horas semanales, hardware estándar (CPU 2 GHz, 8 GB RAM, 256 GB almacenamiento), y pruebas en entornos local (SQLite/MongoDB) y Heroku/Render. Los criterios de aceptación incluyen una aplicación funcional con los módulos mencionados, interfaz responsiva, tiempo de respuesta <2s, uptime >99%, informe APA completo, y tablero Trello actualizado. Un formulario en Google Forms se usará para el levantamiento de información con la familia Aldana Gómez.

Estructura del Desglose del Trabajo (EDT)

Planeación

- **1.1.** Redactar contextualización de la necesidad
- **1.2.** Definir planteamiento del problema

- **1.3.** Establecer objetivo general y específicos
- **1.4.** Redactar alcance, restricciones y criterios de aceptación
- **1.5.** Definir metodología ágil Kanban y configurar tablero Trello
- **1.6.** Crear matriz de riesgos
- **1.7.** Estimar presupuesto

Análisis

- **2.1.** Realizar levantamiento de información (herramientas y conclusiones)
- **2.2.** Mapear stakeholders y clasificarlos
- **2.3.** Definir requerimientos funcionales (RQF) y no funcionales (RQNF)
- **2.4.** Redactar historias de usuario
- **2.5.** Diseñar diagrama de flujo de solución
- **2.6.** Configurar repositorio GitHub (rama main, ramas por funcionalidad)

Diseño

- **3.1.** Diseñar arquitectura del sistema (backend, frontend, base de datos)
- **3.2.** Diseñar base de datos (tablas en SQLite/MongoDB)
- **3.3.** Diseñar interfaz de usuario (wireframes en Figma o papel)
- **3.4.** Modelar sistema (diagramas UML)

Desarrollo

- **4.1.** Configurar entorno en Visual Studio Code (extensiones Python, Pylance, GitLens, Live Server)
- **4.2.** Desarrollar módulo de autenticación (Flask-Login)

- **4.3.** Desarrollar módulo de registro de ingresos y gastos y de visualización (tablas y gráficos con Chart.js)

Pruebas

- **5.1.** Realizar pruebas unitarias (pytest)
- **5.2.** Realizar pruebas funcionales (manuales)
- **5.4.** Documentar resultados de pruebas

Despliegue

- **6.1.** Configurar despliegue en Render o Heroku (integración con GitHub)
- **6.2.** Verificar accesibilidad y seguridad de la aplicación desplegada

Documentación

- **7.1.** Redactar informe académico en PDF (incluyendo todos los entregables)
- **7.2.** Actualizar README.md en GitHub
- **7.3.** Capturar evidencias de Trello y pruebas

Objetivos

General

Diseñar e implementar una aplicación web que centralice el registro, monitoreo y análisis de los ingresos y gastos familiares, identificando a los responsables y ofreciendo herramientas de visualización para prevenir desequilibrios financieros y apoyar una toma de decisiones informada en el hogar.

Específicos

- Analizar la situación actual de manejo de gastos familiares y definir los requerimientos funcionales y técnicos de la aplicación web.
- Diseñar la arquitectura, la base de datos y la interfaz de usuario de la aplicación web, asegurando claridad, usabilidad y seguridad.
- Desarrollar e integrar módulos principales del sistema, el acceso al integrante de la familia, registro de ingresos, gastos y reportes gráficos de patrones de consumo.
- Validar y desplegar la aplicación mediante pruebas unitarias y de funcionamiento, documentando el proceso y garantizando su accesibilidad en un entorno web.

Metodología ágil seleccionada

El proyecto WalletMoneyFamily implementará la metodología ágil Kanban en Trello para gestionar el desarrollo de una aplicación web que permita a una familia organizar y centralizar el registro, monitoreo y análisis de sus ingresos y gastos, identificando quién genera cada ingreso y quién realiza cada gasto. El tablero Trello organiza las tareas en tres listas principales: To Do, In Progress, y Done, alineadas con las fases del SDLC (Planeación, Análisis, Diseño, Desarrollo, Pruebas, Despliegue, Documentación). Cada tarjeta representa una tarea específica, con descripción, responsable (estudiante), periodo estimado. La metodología ágil Kanban en Trello permite priorizar tareas, visualizar el progreso, y ajustar el flujo, garantizando la entrega en 6-8 semanas. Enlace del tablero Trello:

<https://trello.com/b/hUP9ODRf/walletmoneyfamily> .

Justificación del Alcance

Corto Plazo (MVP, 6-8 semanas): El alcance se centra en un prototipo funcional con módulos básicos de autenticación, registro de ingresos/gastos, y visualización de reportes, resolviendo el problema principal de desorganización financiera en la familia Aldana Gómez.

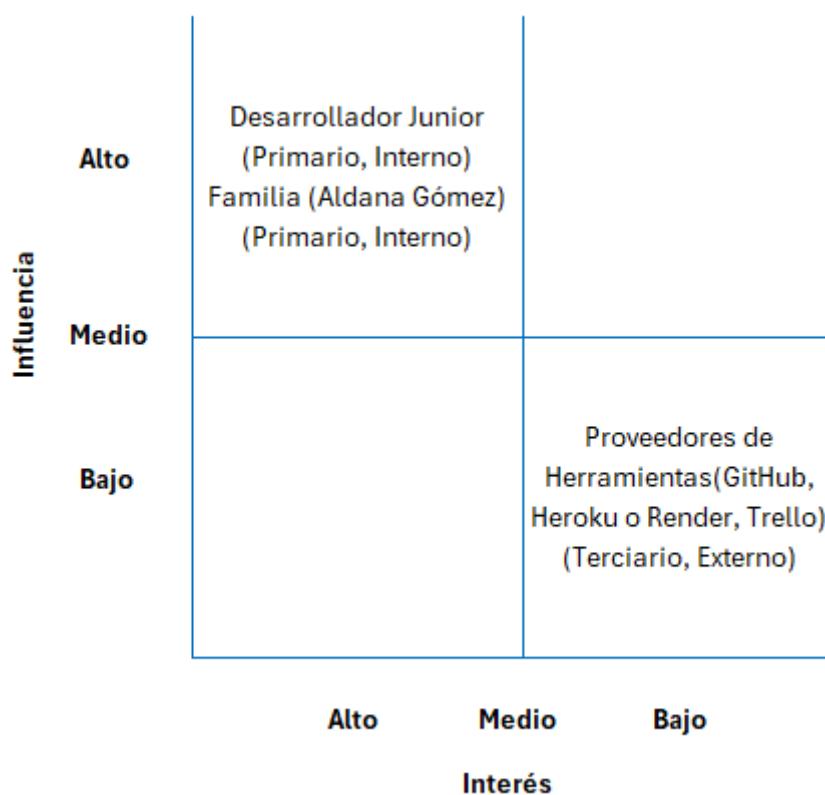
Esta solución inicial permite registrar y analizar datos para evitar desequilibrios inmediatos, respondiendo a stakeholders.

Medio Plazo (3-6 meses post-desarrollo): Se podría expandir con funcionalidades como exportación de reportes a PDF o filtros avanzados para patrones de gasto, mejorando la toma de decisiones informada. Esto respondería a stakeholders al proporcionar mayor transparencia, pero se limita al MVP actual por restricciones de tiempo y presupuesto.

Largo Plazo (6-12 meses): Se podría integrar con apps de productividad o agregar usuarios múltiples para familias extendidas, promoviendo escalabilidad. Esta descripción de soluciones aborda el problema de la gestión desorganizada, ofreciendo a stakeholders (familia, estudiante, profesores) una herramienta que centraliza finanzas reduce conflictos, y facilita decisiones informadas, contribuyendo a un equilibrio financiero sostenido.

Figura 1

Mapa de Stakeholders (Influencia vs Interés)



Fuente: Elaboración propia.

Tabla 1

Matriz de Riesgos

Riesgo	Probabilidad	Impacto	Mitigación
Retrasos por imprevistos (e.g., problemas técnicos)	Media	Alto	Hitos semanales en Kanban, en Trello para mantener foco
Errores en lógica de datos (e.g., guardado de ingresos)	Media	Alto	Pruebas unitarias exhaustivas con pytest
Problemas en despliegue (e.g., configuración en Render)	Baja	Medio	Pruebas previas locales y CI/CD con GitHub
Limitaciones de hardware (e.g., rendimiento bajo)	Baja	Medio	Usar herramientas ligeras como Flask y SQLite/MongoDB

Fuente: Elaboración propia

Tabla 2

Presupuesto para el cliente

Concepto	Horas	Costo por hora (USD)	Costo total (USD)	Referencia
Planeación y análisis	15	10	150	Estimación estándar
Diseño (arquitectura, DB, UI)	20	12	240	Estimación estándar
Desarrollo (módulos)	50	20	1000	Estimación estándar
Pruebas (unitarias, funcionales)	10	12	120	Estimación estándar
Despliegue	8	10	80	Estimación estándar
Documentación	10	10	100	Estimación estándar
Total sin contingencia	113		1690	
Margen de contingencia (10%)			169	Estimación estándar
Total estimado	113		1859	

Nota. El proyecto tendrá una duración estimada de 113 horas, distribuidas entre las fases de planeación y análisis, diseño, desarrollo, pruebas, despliegue y documentación, buscando optimizar el uso de recursos en cada etapa. Las tarifas por hora se establecerán en un rango de

10–20 USD/hora, de acuerdo con los valores promedio proyectados para desarrolladores junior. La contingencia se fijará en 10% (169 USD), con el fin de mantener el presupuesto dentro del objetivo aproximado de 1900 USD, estimándose un costo final de 1859 USD.

Fuente: Elaboración propia.

Repositorio

El documento y el código se subirán al repositorio GitHub:

<https://github.com/rfalgo/WalletMoneyFamily> La rama principal es main, y hay ramas por integrante (e.g. se puede usar feature/authenticacion).

Fase de Análisis del ciclo de vida del desarrollo de software (Fase de Planeación de Proyectos)

Levantamiento de Información (Herramientas y Conclusiones)

Herramientas Usadas:

- **Encuestas informales:** Preguntas a miembros de la familia Aldana Gómez sobre su gestión financiera (e.g., "¿Cómo registran ingresos/gastos?").
- **Instrumento:** Formulario en Google Forms con 9 preguntas sobre gestión financiera
Enlace:

https://docs.google.com/forms/d/e/1FAIpQLSePrMfHZlpMrqqAa_qcT5uU3aeV8gBv3GnT09vqQWQtLFFfQTw/viewform?usp=header

Tabla 3

Conclusiones

Conclusión	Evidencia de la Clave	Implicación para el Proyecto
Alta Tasa de Conflicto	Encuesta	
El 83.3% reporta tener conflictos por falta de control financiero con frecuencia.	La aplicación debe priorizar la transparencia total y ayudar a definir responsabilidades claras para reducir la fricción.	

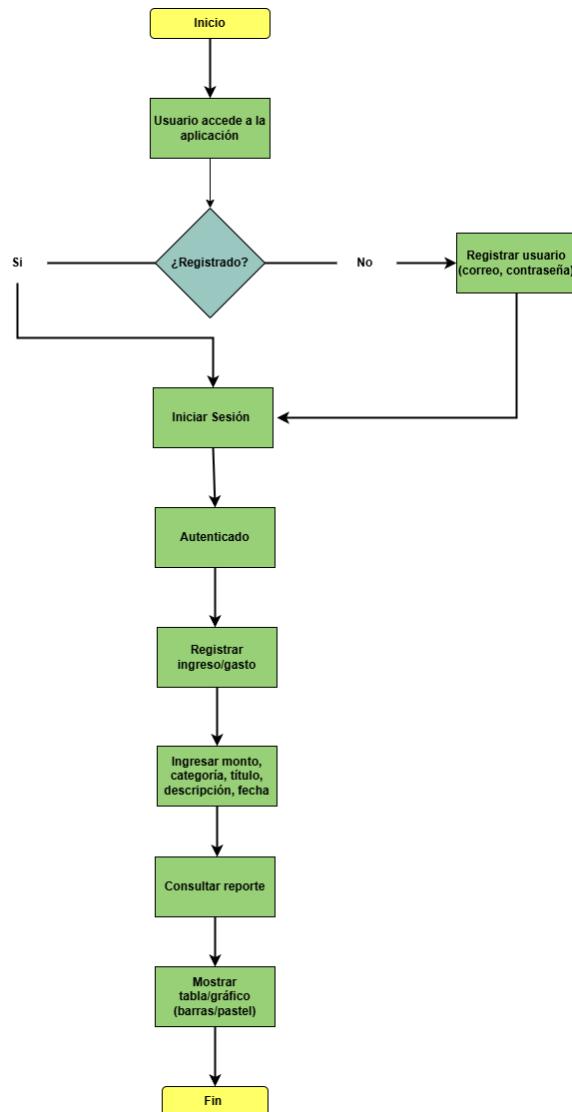
Métodos	El 66.7% usa un cuaderno físico para registrar, y mencionan la pérdida de registros y la falta de disciplina como problemas.	Se justifica la migración total a una plataforma digital que simplifique y centralice el registro, actuando como un único punto de verdad financiero.
Aceptación	El 100% considera la aplicación web	No existe barrera de entrada tecnológica; el foco debe estar en la funcionalidad y usabilidad.
Tecnológica		
Total	totalmente útil y el 100% se siente muy cómodo usando plataformas en línea.	

Necesidad de Claridad y Control	Las funcionalidades más deseadas (100%) son los Reportes gráficos/estadísticos, la Clasificación automática de gastos, y el Registro de ingresos/gastos por persona.	El registro debe ser detallado y por individuo, y la presentación de datos (gráficos) es vital para lograr el beneficio más buscado: Control y mejor planificación.
--	--	---

Nota. La familia Aldana Gómez requiere urgentemente una aplicación web centralizada que registre y analice sus finanzas con visualizaciones gráficas para reducir los conflictos y mejorar significativamente las decisiones de ahorro y planificación. **Fuente:** Elaboración propia.

Figura 2

Diagrama de Flujo de Solución Seleccionada



Fuente: Elaboración propia.

Historias de usuario

Tabla 4

Historias de Usuario

ID	Historia	Descripción	Criterios de aceptación	Puntos de esfuerzo
----	----------	-------------	-------------------------	--------------------

HU1	Registro/Inicio de sesión	Como usuario, quiero registrarme e iniciar sesión para gestionar mis finanzas.	Formulario de registro e inicio de sesión funcional; contraseñas con hash; redirección a dashboard.	5
HU2	Registro de ingresos/gastos	Como usuario, quiero registrar ingresos/gastos con monto, categoría, título, descripción, fecha.	Formulario guarda datos en SQLite; valida campos; confirma registro del miembro de la familia.	5
HU3	Visualización de reportes	Como usuario, quiero ver tablas y gráficos de ingresos/gastos por categoría/miembro.	Tablas y gráficos (barras/pastel) generados con Chart.js; filtros por periodo.	5

Fuente: Elaboración propia

Requerimientos funcionales (RQF) y no funcionales (RQNF)

Tabla 5

Requerimientos funcionales (RQF)

ID	Nombre	Descripción	Usuarios	Prioridad
RQF1	Autenticación	Registro e inicio de sesión con correo y contraseña (hash con Flask-Login).	Miembros de la familia	Alta
RQF2	Registro de ingresos/gastos	Registrar ingresos (monto, categoría, título, descripción, fecha) y gastos (monto, categoría, título, descripción, fecha) en SQLite.	Miembros con permisos	Alta
RQF3	Generación de reportes	Mostrar tablas y gráficos (barras/pastel) de ingresos/gastos por miembro/categoría.	Todos los miembros	Alta

Fuente: Elaboración propia.

Tabla 6

Requerimientos no funcionales (RQNF)

ID	Nombre	Descripción	Prioridad
RQNF1	Tiempo de respuesta	Respuesta <2 segundos para consultas y registros.	Alta

RQNF2	Interfaz responsiva	Compatible con móviles y desktops (Bootstrap).	Media
RQNF3	Uptime	>99% en Render.	Alta
RQNF4	Seguridad	Contraseñas con hash; protección contra accesos no autorizados.	Alta

Fuente: Elaboración propia.

Fase de Diseño del ciclo de vida del desarrollo de software (Fase de Ejecución de Proyectos)

Prototipos de interfaz

Prototipo de Baja Fidelidad

Enlace: <https://balsamiq.cloud/srw8uws/py4692s/r2278>

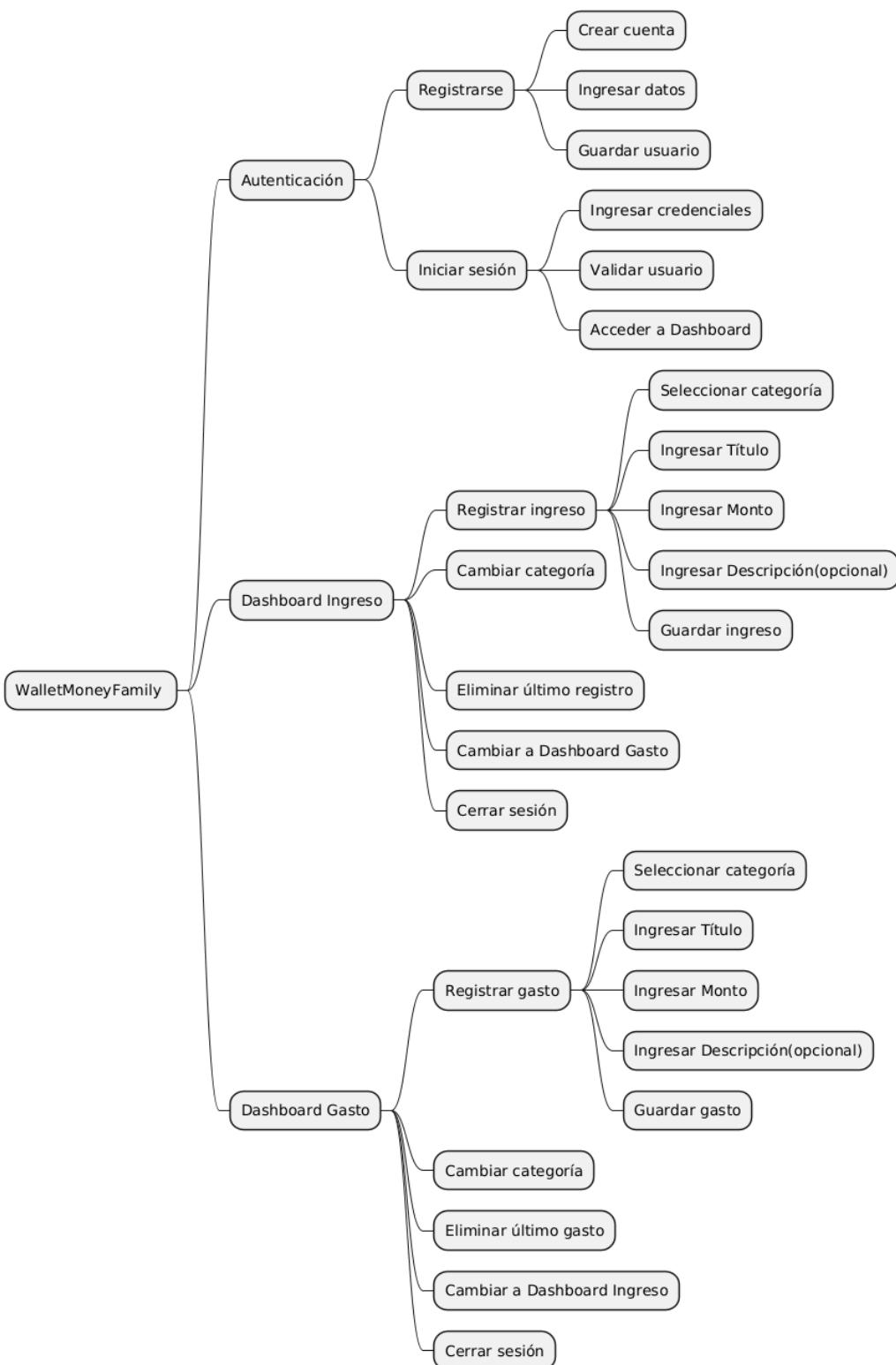
Prototipo de Alta Fidelidad

Enlace:

<https://www.figma.com/design/N36uw8AP7PaxZ1n5Et9pXS/WalletMoneyFamily?node-id=1-845&t=Md3g6x6yCMocePpu-1>

Figura 3

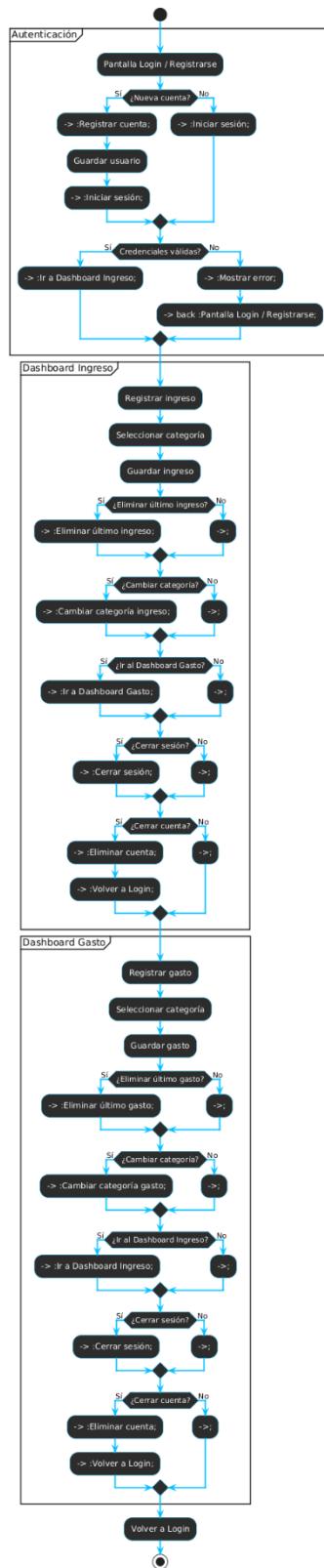
User Map



Fuente: Elaboración propia.

Figura 4

User Flow



Fuente: Elaboración propia.

Modelos de comportamiento y estructura

Tabla 7

Casos de Uso Documentados

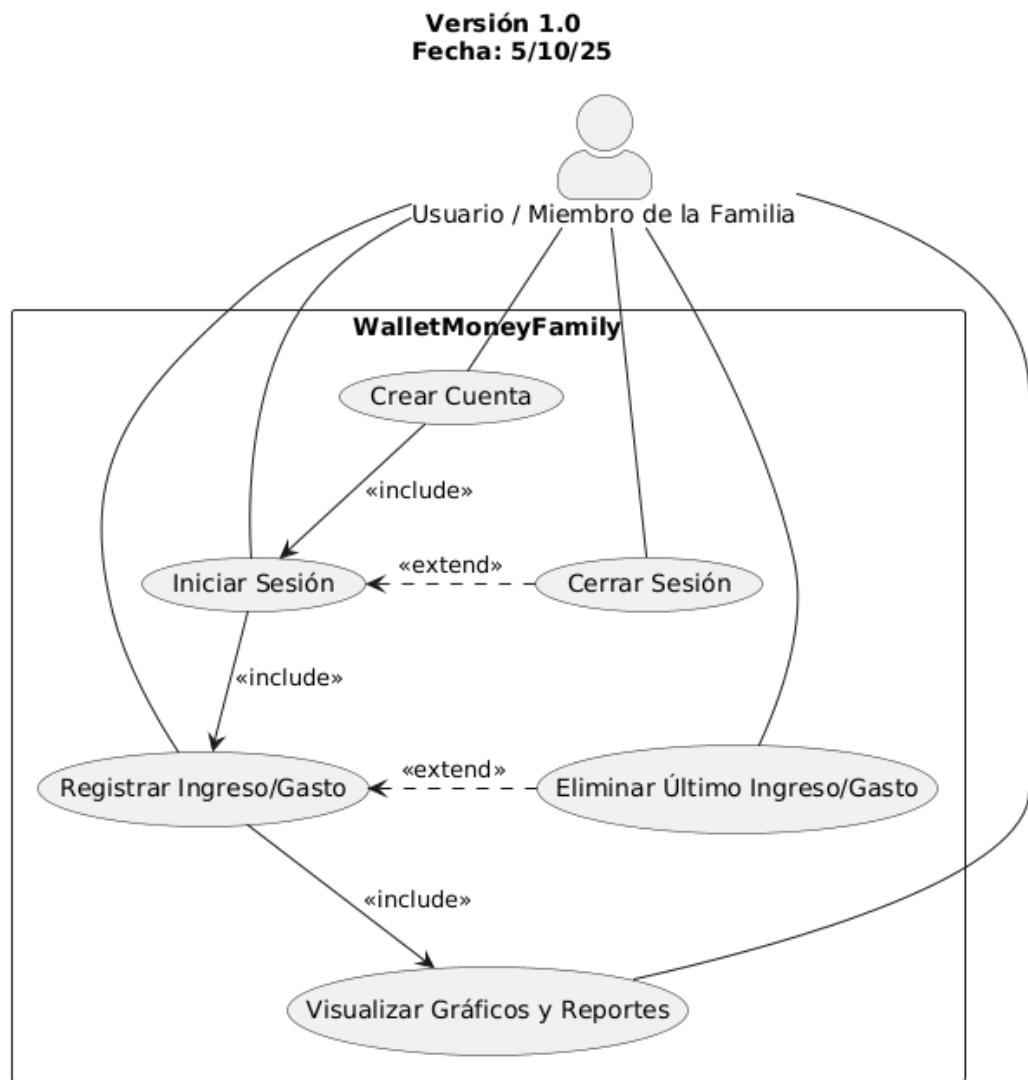
ID	Nombre	Descripción	Actores	Precondiciones	Flujo Normal	Flujos Alternativos/Exceptionales	Postcondiciones	Frecuencia
CU1	Crear Cuenta	Registrar nuevo usuario y unirse o crear familia	Nuevo usuario	Acceso a internet, disponer de un correo electrónico	1. Accede a login 2. Selecciona "Crear Cuenta" 3. Ingresar datos 4. Guarda	Excepcional: Correo duplicado → error.	Cuenta creada, redirige a login	Alta
CU2	Iniciar Sesión	Autenticar al usuario con correo y contraseña para acceder al sistema.	Miembro de la familia	Usuario registrado con credenciales válidas.	1. Accede a la página de login. 2. Ingresar correo y contraseña. 3. Sistema valida credenciales. 4. Redirige al dashboard.	Excepcional: Credenciales inválidas → muestra error.	Usuario autenticado y en el dashboard.	Alta
CU3	Cerrar Sesión	Finalizar la sesión del usuario autenticado en el sistema.	Miembro autenticado	Usuario con sesión activa	1. Usuario selecciona la opción “Cerrar Sesión”. 2. El sistema limpia los datos temporales. 3. Redirige a la pantalla de inicio.	Excepcional: Error de conexión → mantiene sesión temporal.	Sesión finalizada; usuario desconectado.	Media

CU4	Registrar Ingreso/Gasto	Registrar un ingreso o gasto con nombre, monto, miembro, categoría y descripción.	Miembro autenticado	Usuario logueado; formulario disponible.	<ol style="list-style-type: none"> 1. Accede al formulario de ingreso/gasto. 2. Ingresa datos válidos y guarda. 3. Confirma registro. 	Alternativo: Registro rápido (solo monto y categoría). Excepcional: Datos inválidos → muestra error.	Ingreso/gasto registrado.	Alta
CU5	Eliminar Último Ingreso/Gasto	Permitir al usuario eliminar el último ingreso o gasto registrado.	Miembro autenticado	Usuario logueado; existe al menos un registro.	<ol style="list-style-type: none"> 1. Usuario selecciona “Eliminar último”. 2. Sistema solicita confirmación. 3. Elimina el registro y actualiza reportes. 	Excepcional: No hay registros disponibles → muestra mensaje “Sin datos para eliminar”.	Registro eliminado correctamente.	Media
CU6	Visualizar Gráficos y Reportes	Visualizar tablas y gráficos por miembro o categoría.	Miembro autenticado	Datos registrados.	<ol style="list-style-type: none"> 1. Sistema genera reporte. 2. Visualización de reportes. 3. Muestra tablas/gráficos y resumen. 	Excepcional: Sin datos → muestra “No hay datos”.	Reportes visualizados.	Alta

Fuente: Elaboración propia.

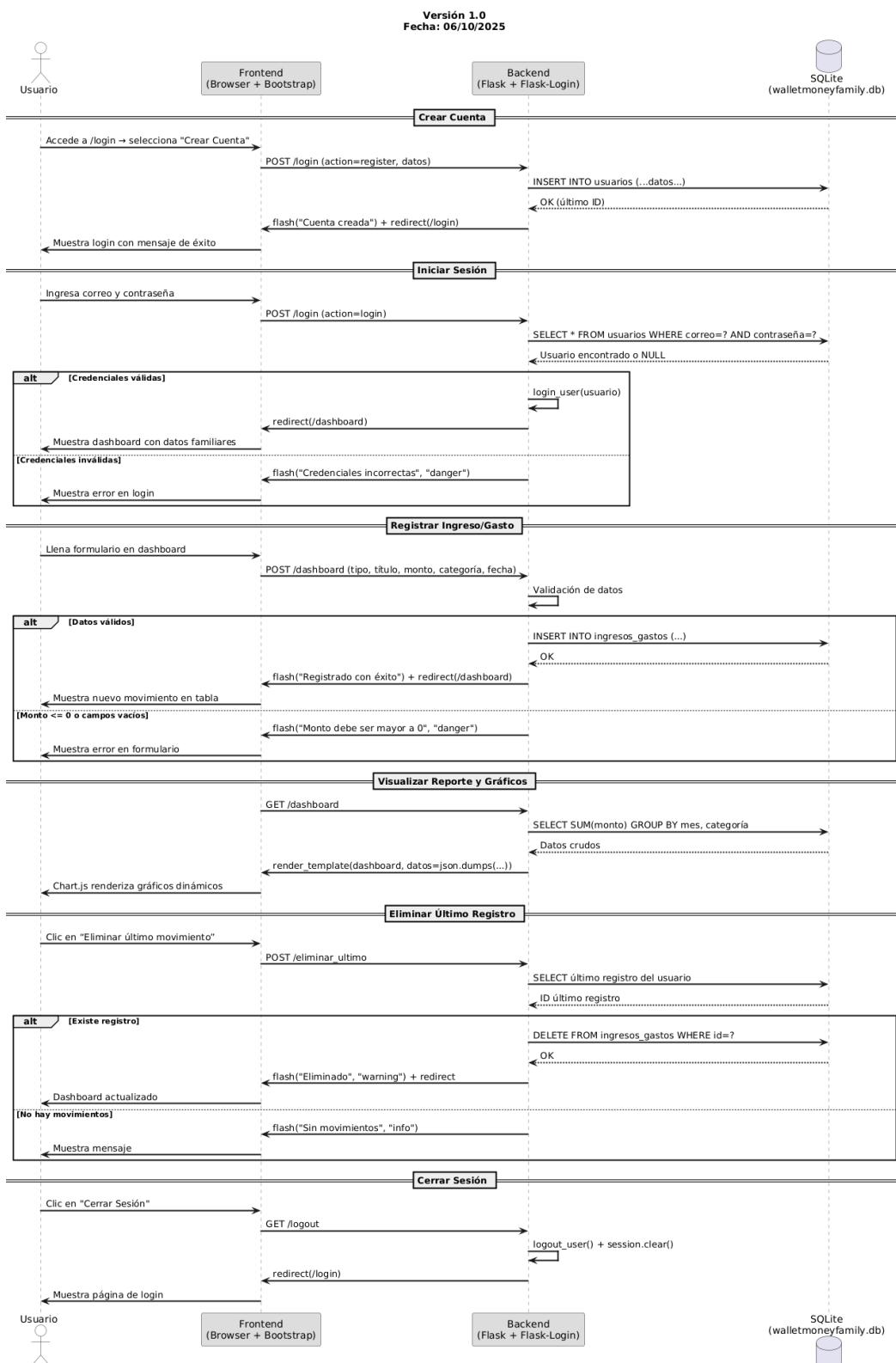
Figura 5

Diagrama Casos de Uso



Fuente: Elaboración propia

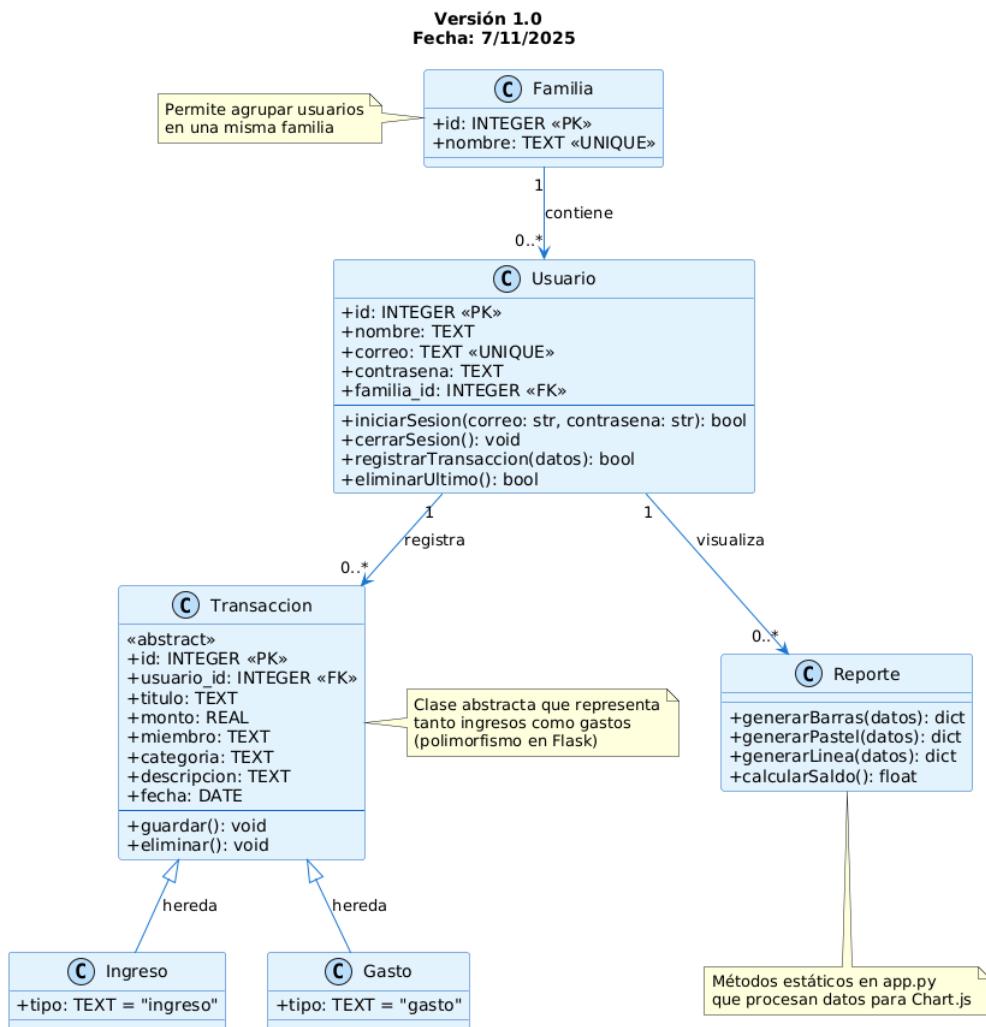
Diagrama de Secuencia



Fuente: Elaboración propia.

Figura 7

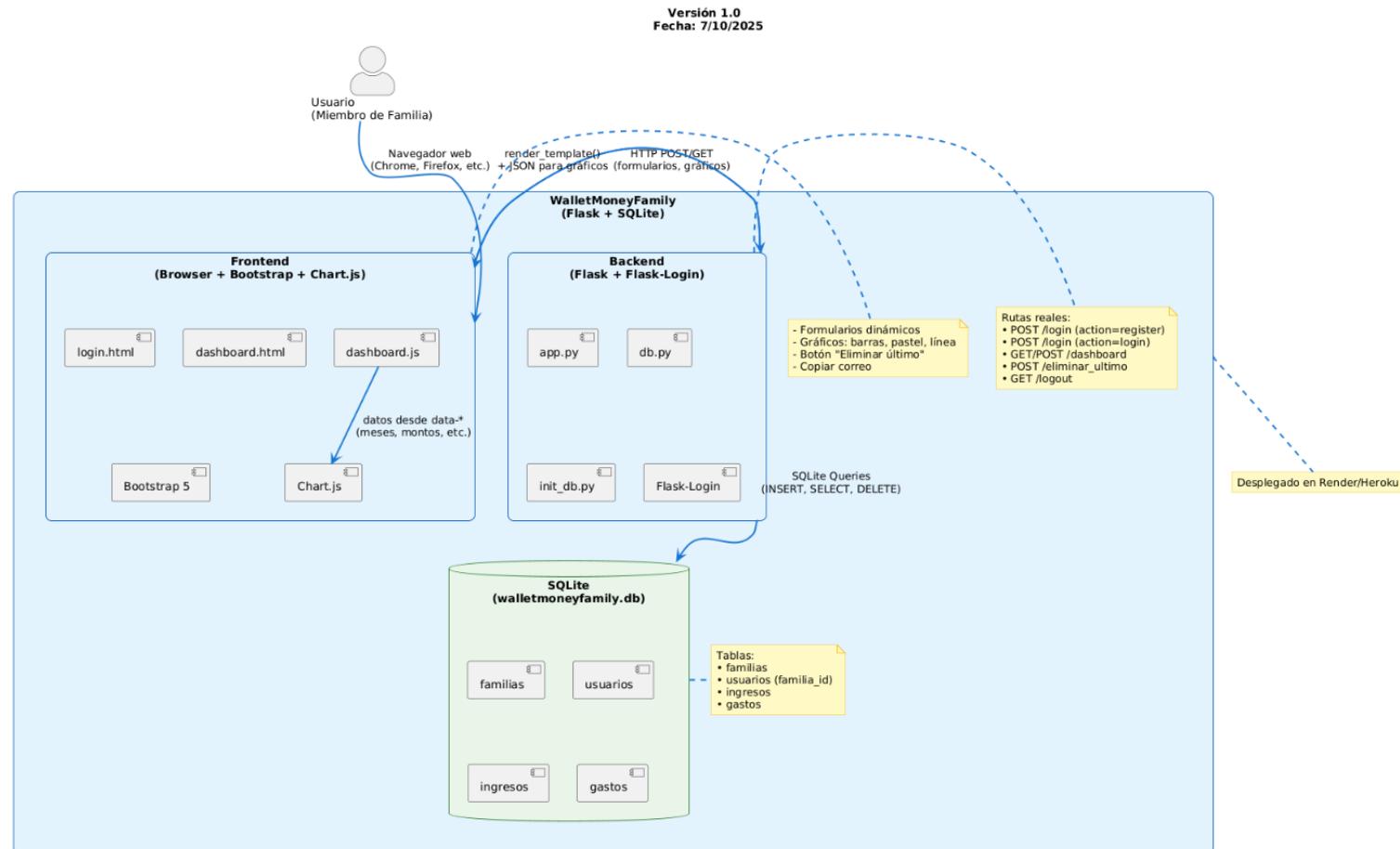
Diagrama de clases



Fuente: Elaboración propia.

Figura 8

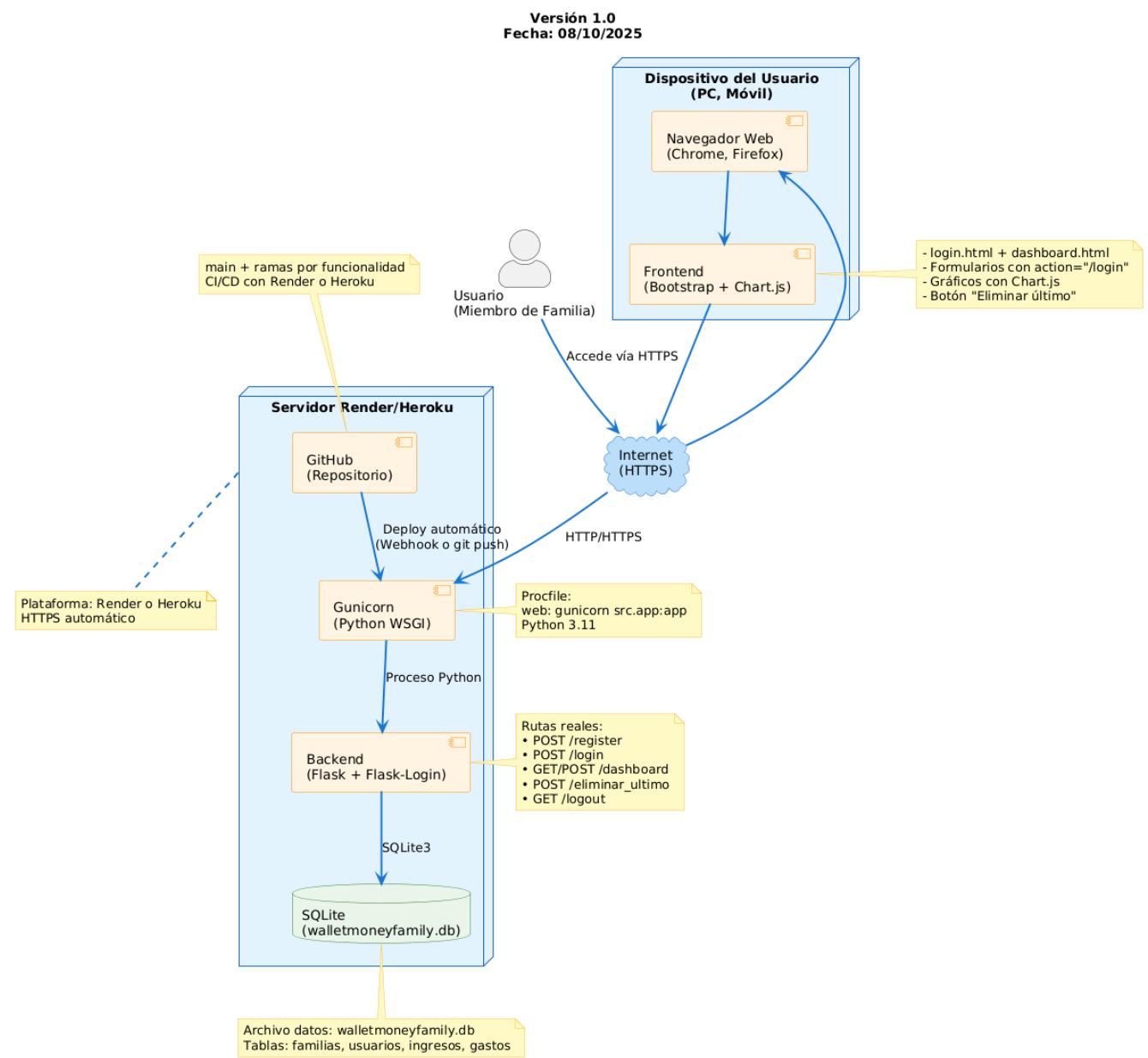
Diagrama de Componentes



Fuente: Elaboración propia.

Figura 9

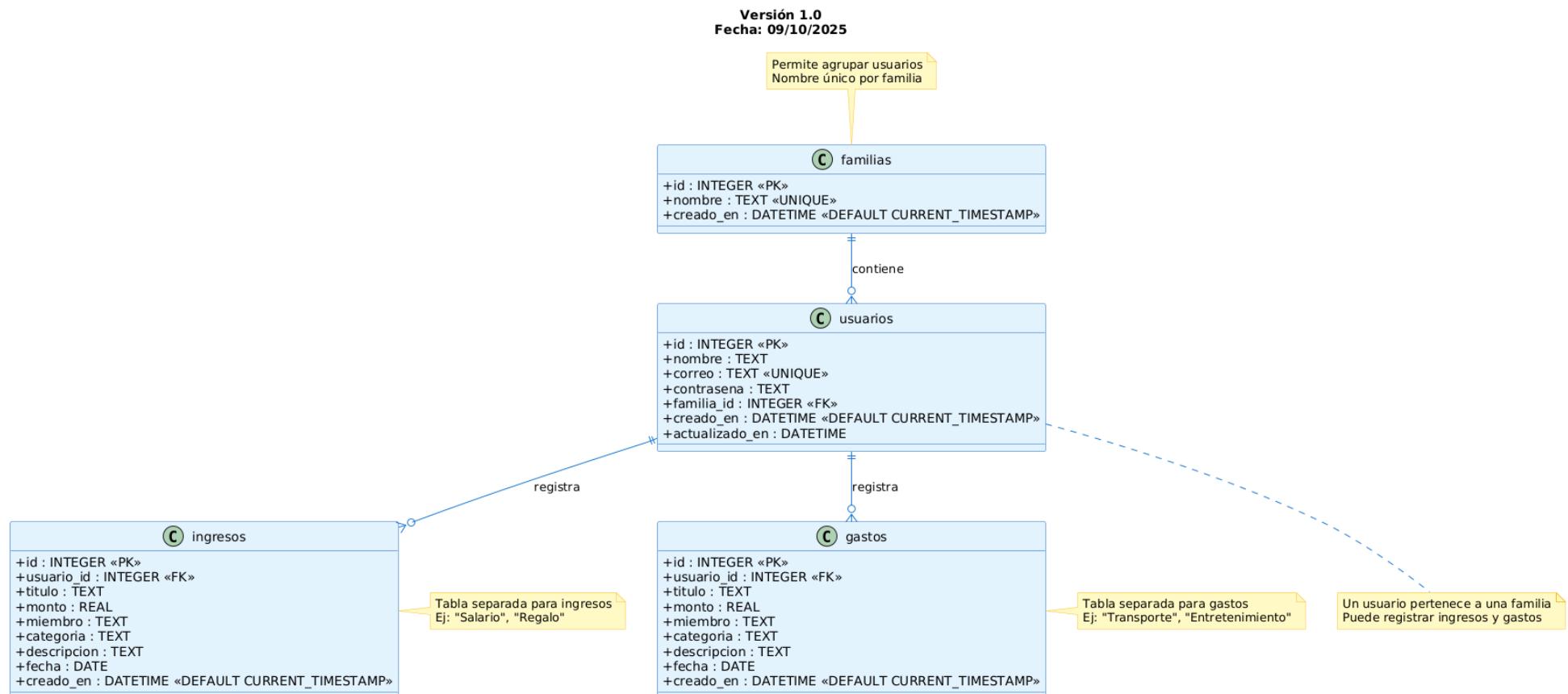
Diagrama de Despliegue



Fuente: Elaboración propia.

Figura 10

Diagrama Relacional



Fuente: Elaboración propia.

Pruebas Usabilidad

Proyecto: WalletMoneyFamily

Versión del prototipo: 1.0

Fecha de la prueba: 14 de noviembre de 2025

Equipo responsable: Desarrollador Tester

Herramientas usadas: Figma (prototipo alta fidelidad) + Maze.co

Objetivos de la prueba:

- Validar la facilidad de uso del flujo principal
- Identificar puntos de fricción en registro, transacciones y cierre
- Recolectar retroalimentación cualitativa

Número de participantes: 8 usuarios reales (18–35 años, estudiantes y profesionales)

Tareas / Escenarios: 7 (superan el mínimo de 5)

Enlace del reporte: <https://app.maze.co/report/Pruebas-de-usabilidad-WalletMoneyFamily/awgz7mhz1gxd0/intro>

Tabla 8

Tareas Realizadas

#	Tarea	Tiempo Promedio	Tasa de Éxito
1	Iniciar la aplicación y crear una cuenta	7.3s	100%
2	Iniciar sesión	5.0s	100%
3	Registrar un ingreso	11.1s	100%
4	Cambiar de ingreso a gasto	4.7s	100%
5	Registrar un gasto	5.1s	100%
6	Eliminar el último registro	4.6s	100%
7	Cerrar sesión	4.1s	100%

Fuente: Elaboración propia

Tasa de éxito global: 100%

Tiempo promedio por tarea: 6.1 segundos

Tasa de abandono (drop-off): 0%

Tabla 9

Resultados Cuantitativos

Participantes completados	8 / 8
Tiempo total promedio	42.9 segundos
Tasa de misclicks	0% (excepto 13% en ingreso – se optimizará)
Calificación promedio (1–10)	9.7 / 10

Fuente: Elaboración propia.

Figura 11

Comentarios cualitativos (usuarios reales)

Responses 

Me gusta la experiencia, la aplicación tiene una gran navegabilidad  Participant 469191571	POSITIVE
Me gusto mucho la experiencia fue muy agradable el uso de esta aplicación  Participant 467561469	POSITIVE
Fue una gran experiencia estas pruebas de usabilidad  Participant 46984165	POSITIVE
Muy buena la experiencia  Participant 469171359	POSITIVE
AAGRADABLE  Participant 469157453	NEUTRAL
Me gusta mucho la navegabilidad de la aplicación  Participant 469160728	POSITIVE
Me pareció una experiencia agradable, una aplicación intuitiva de manejar  Participant 469139737	POSITIVE

Fuente: Elaboración propia

Evidencias Adjuntas

- PDF oficial de Maze (8 usuarios): <https://drive.google.com/file/d/1ge9ac6PZwg0Vz28lrUDZavtYunFUXj9/view?usp=sharing>
- Enlace público Maze: <https://t.maze.co/468354761>
- Prototipo Figma usado:
<https://www.figma.com/design/N36uw8AP7PaxZ1n5Et9pXS/WalletMoneyFamily?node-id=1-845&t=Md3g6x6yCMocePpu-1>

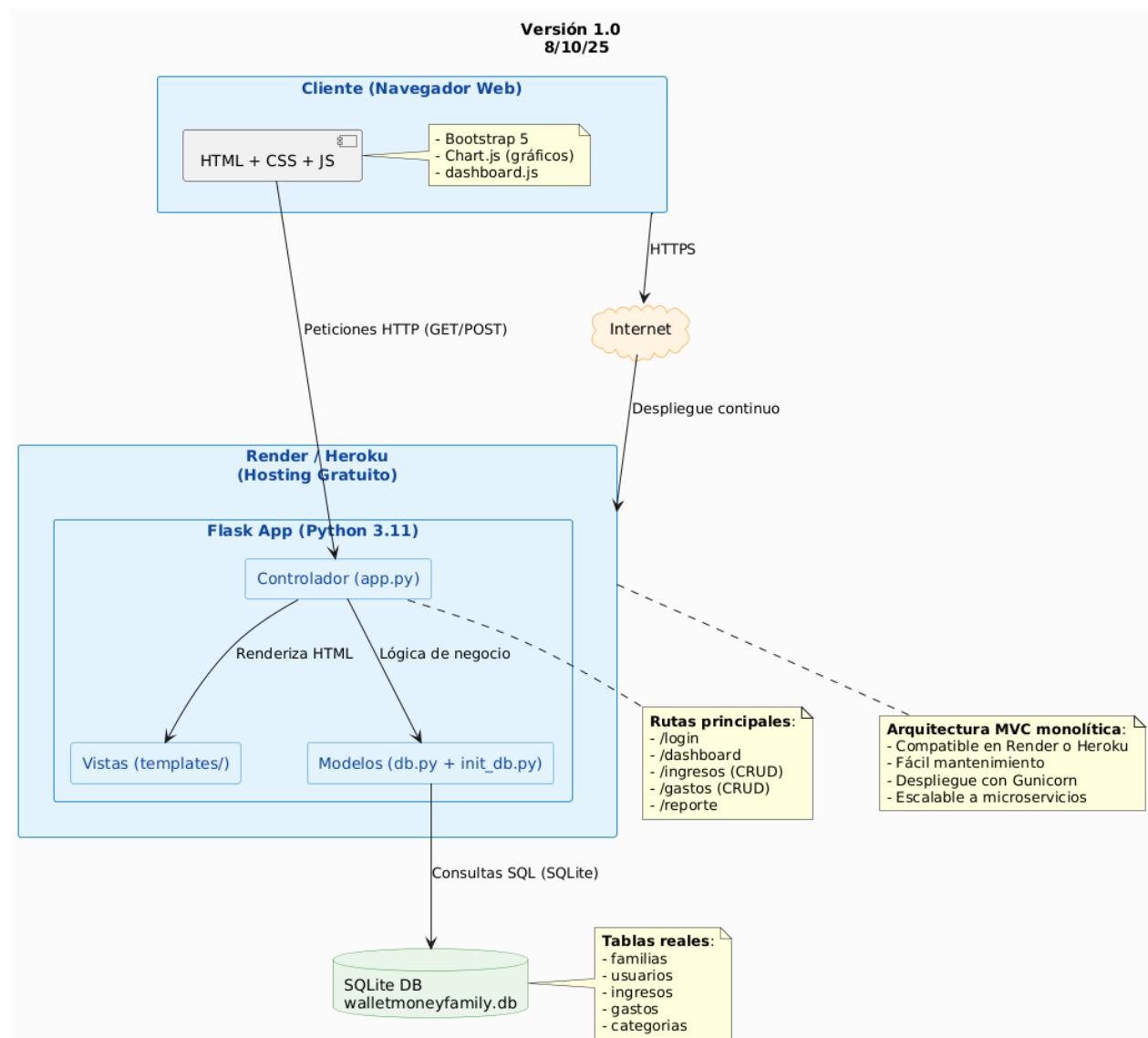
Conclusión:

El prototipo es altamente usable, con 100% de éxito en todas las tareas y una calificación de 9.7/10.

Fase de Desarrollo del ciclo de vida del desarrollo de software (Fase de Ejecución de Proyectos)

Figura 12

Diagrama de Arquitectura



Fuente: Elaboración propia.

Prototipo Funcional

Tabla 10

Patrones Arquitectónicos y de Diseño Aplicados

Patrón	Capa	Descripción
MVC (Model-View-Controller)	General	
	Model	walletmoneyfamily.db + consultas SQL en app.py
	View	templates/*.html + Bootstrap
	Controller	Rutas y lógica en app.py
Flask Blueprint	Backend	Organización modular
Template Inheritance	Frontend	login.html, dashboard.html
Static File Serving	Frontend	/static/css, /static/js, /static/images
SQLite Local +	Persistencia	Desarrollo local
Bootstrap Responsive	UI	Diseño adaptable a móvil y web

Fuente: Elaboración propia

Figura 13

Estructura del directorio del proyecto

```
PS C:\Users\robal\Downloads\WalletMoneyFamily> tree /F
Listado de rutas de carpetas
El número de serie del volumen es D282-426D
C:.
    app.py
    init_db.py
    requirements.txt
    walletmoneyfamily.db

    docs
        evidencias
            diagramas
                Diagrama Casos de Uso.png
                Diagrama de Arquitectura.png
                Diagrama de clases.png
                Diagrama de Componentes.png
                Diagrama de Despliegue.png
                Diagrama de Flujo de Solución Seleccionada.png
                Diagrama de Secuencia.png
                Diagrama Relacional.png
                Mapa de Stakeholders (Influencia vs Interés).png
                User Flow.png
                User Map.png

            formularios
                Evidencia 1.png
                Evidencia 2.png
                Evidencia 3.png
                Evidencia 4.png
                Evidencia 5.png

            pruebas de usabilidad
                Maze Pruebas de usabilidad WalletMoneyFamily.pdf

        trello
            Evidencia 1 Trello.png
            Evidencia 2 Trello.png

        informe
            Primer Avance Informe APA .pdf
            Segundo Avance Informe APA.pdf

    src
        db.py

        static
            css
                bootstrap.min.css

            images
                icon dashboard.png
                icon1.png

            js
                dashboard.js

        templates
            dashboard.html
            login.html

PS C:\Users\robal\Downloads\WalletMoneyFamily> []
```

Fuente: Elaboración propia.

Tabla 11*Avances de Codificación*

Módulo	Estado	%	Comentario
Completado			
app.py – Rutas básicas	OK	100%	/login, /dashboard, /
Login HTML + Form	OK	100%	Bootstrap + validación JS
Dashboard HTML + JS	OK	100%	Gráficos con Chart.js
Base de datos SQLite	OK	100%	Tablas: usuarios, movimientos, categorías
Registro de ingresos/gastos	OK	100%	Formulario funcional
Eliminar último movimiento	OK	100%	Botón presente y lógica

Fuente: Elaboración propia**Enlace de Repositorio GitHub**<https://github.com/rfalgo/WalletMoneyFamily.git>

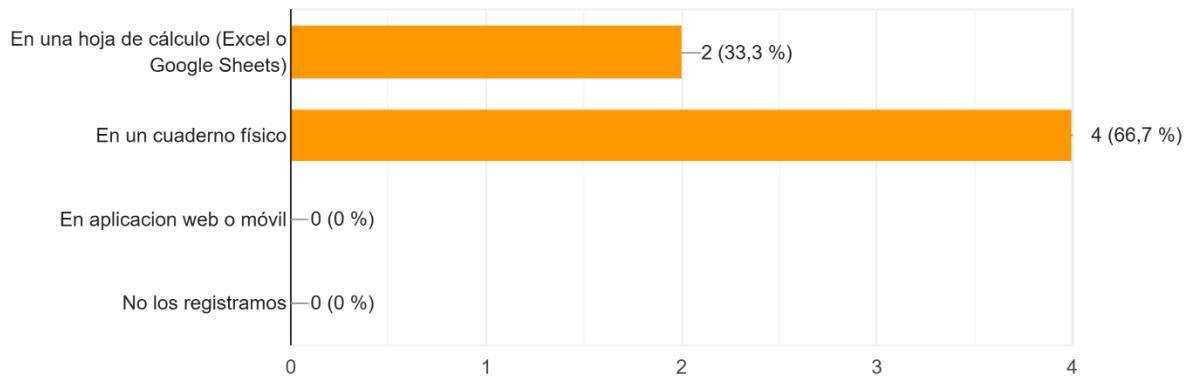
Anexos

Figura 14

Formulario pregunta 1

¿Cómo registran actualmente sus ingresos y gastos?

6 respuestas



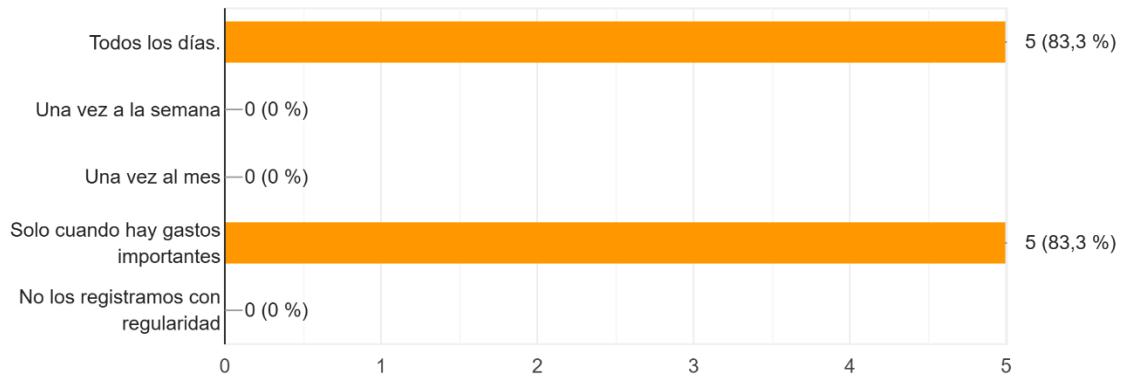
Fuente: Elaboración propia con Google Forms.

Figura 15

Formulario pregunta 2

¿Con qué frecuencia registran sus gastos?

6 respuestas



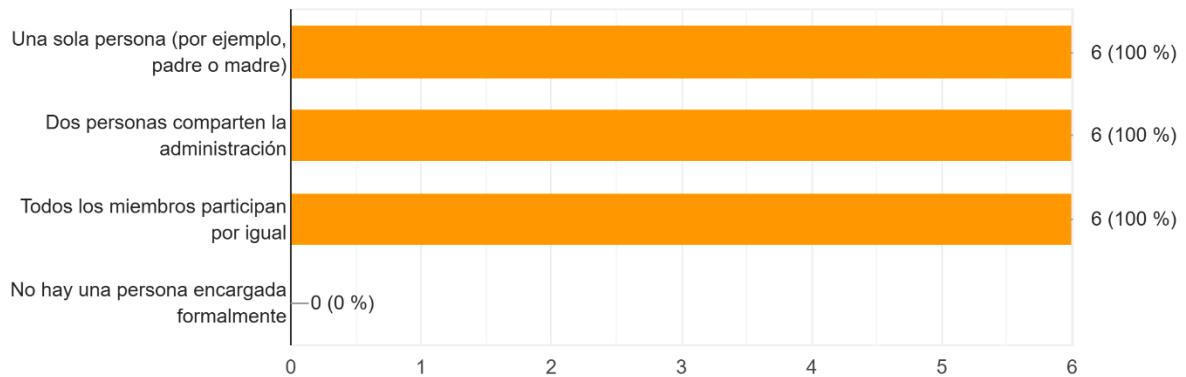
Fuente: Elaboración propia con Google Forms.

Figura 16

Formulario pregunta 3

¿Quién administra los ingresos del hogar?

6 respuestas



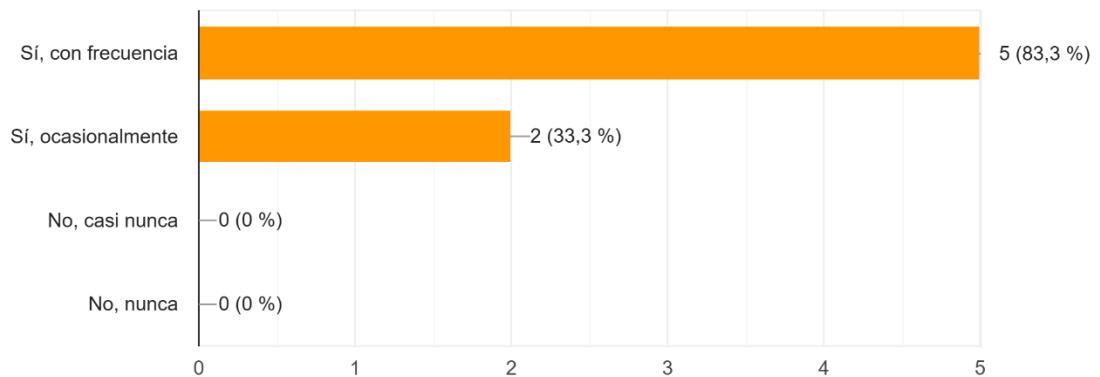
Fuente: Elaboración propia con Google Forms.

Figura 17

Formulario pregunta 4

¿Han tenido conflictos o discusiones por falta de control financiero?

6 respuestas



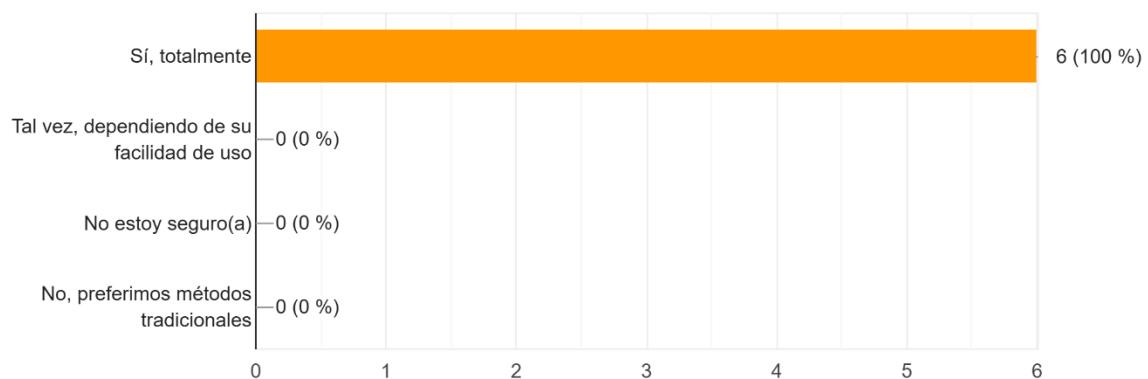
Fuente: Elaboración propia con Google Forms.

Figura 18

Formulario pregunta 5

¿Consideran útil una aplicación web que centralice el registro y análisis de finanzas familiares?

6 respuestas



Fuente: Elaboración propia con Google Forms.

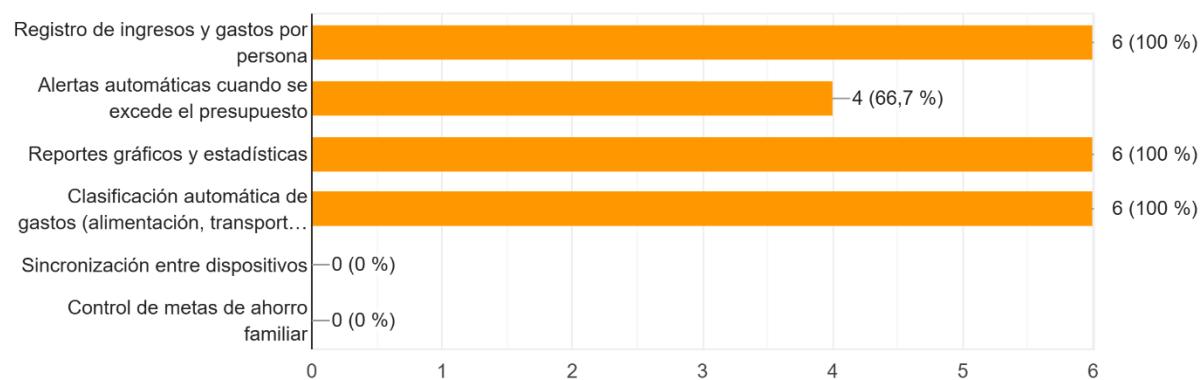
Figura 19

Formulario pregunta 6

¿Qué funcionalidades consideran más importantes para una aplicación web familiar de finanzas?

(Seleccione máximo 3)

6 respuestas



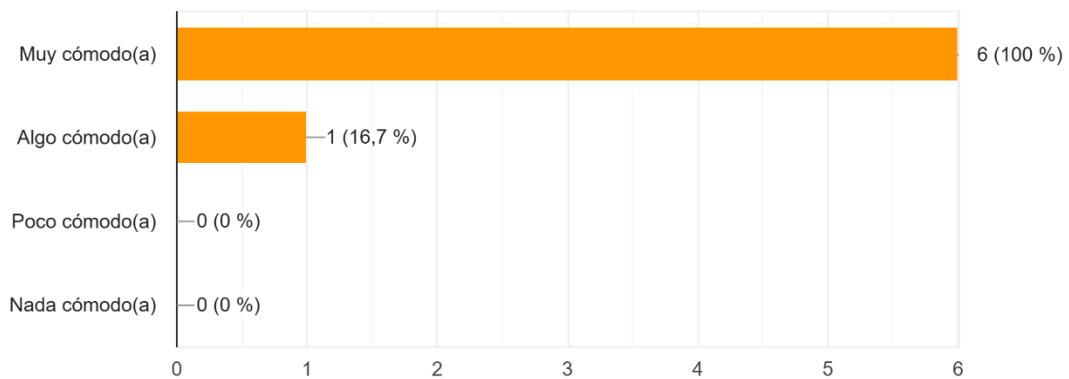
Fuente: Elaboración propia con Google Forms.

Figura 20

Formulario pregunta 7

¿Qué tan cómod(a) se siente usando aplicaciones web o plataformas en línea?

6 respuestas



Fuente: Elaboración propia con Google Forms.

Figura 21

Formulario pregunta 8

¿Qué problemas enfrentan actualmente al registrar sus finanzas?

6 respuestas

Disponer de la información de manera gráfica de gastos e ingresos

Falta de organización

No hay disciplina para anotar

Dificultad para consolidar la información

Falta de tiempo para actualizar

Se pierden los registros

Fuente: Elaboración propia con Google Forms.

Figura 22

Formulario pregunta 9

¿Qué beneficios esperan obtener de una aplicación web de finanzas familiares?

6 respuestas

- Entender como se está manejando los gatos de la familia, y evitar perdidas de dinero
- Control financiero
- Saber en qué gasto
- Evitar conflictos familiares
- Control y organización
- Ahorro y mejor planificación

Fuente: Elaboración propia con Google Forms.

Conclusión

Se espera que el proyecto WalletMoneyFamily entregue un Producto Mínimo Viable (MVP) funcional que abordará la necesidad de la familia Aldana Gómez de organizar la gestión financiera del hogar. Mediante la implementación de una aplicación web con módulos de autenticación y registro/seguimiento de ingresos y gastos, el proyecto cumplirá con su objetivo general de centralizar el registro, monitoreo y análisis de las finanzas familiares. Los objetivos específicos; analizar la situación financiera actual, diseñar la arquitectura y la interfaz del sistema, desarrollar los módulos principales y validar mediante pruebas y despliegue, se lograrán dentro del plazo de 6 a 8 semanas. La metodología ágil Kanban, gestionada en un tablero Trello con columnas To Do, In Progress y Done, asegurará una gestión eficiente de tareas y un seguimiento claro del progreso, alineado con las fases del Ciclo de Vida del Desarrollo de Software (SDLC): Planeación, Análisis, Diseño, Desarrollo, Pruebas, Despliegue y Documentación. La aplicación, que se desarrollará con Flask para el backend, SQLite o MongoDB para la base de datos, Bootstrap para una interfaz responsive y Chart.js para reportes gráficos, cumplirá con los requerimientos funcionales (autenticación, registro de datos y generación de reportes) y no funcionales (tiempo de respuesta <2 segundos, interfaz responsive, uptime >99%, contraseñas con hash), los cuales se validarán

mediante pruebas unitarias con pytest y pruebas funcionales manuales. El presupuesto para la familia debe ser de 1859 USD, mientras que el presupuesto para el desarrollador junior se mantendrá en ~25 USD gracias al uso de herramientas gratuitas o de bajo precio como Flask, SQLite o MongoDB, Trello, GitHub y Render o Heroku. La encuesta en Google Forms orientará el diseño de la aplicación, priorizando la transparencia y los reportes gráficos para garantizar usabilidad y claridad, según las necesidades identificadas de la familia Aldana Gómez. A corto plazo, el MVP resolverá la desorganización financiera inmediata; a mediano plazo (3-6 meses tras el desarrollo), se espera incorporar funcionalidades como exportación de reportes a PDF o filtros avanzados para mejorar la toma de decisiones; a largo plazo (6-12 meses), la integración con aplicaciones de productividad o el soporte para familias extendidas podría potenciar la escalabilidad. WalletMoneyFamily no solo proporcionará a la familia Aldana Gómez una herramienta para reducir conflictos y mejorar la planificación financiera, respaldado por el tablero Trello, el repositorio GitHub y un informe en formato APA, demostrando una gestión efectiva del proyecto y una ejecución técnica sólida.

Referencias Bibliográficas

- Omaña, M. (2012). *Manufactura esbelta: una contribución para el desarrollo de software con calidad*. Red Enlace, pp. 14–18.
- Pressman, R. S. (2021). *Ingeniería de software* (Cap. 24–26, pp. 490–548). McGraw-Hill Interamericana.
- Sommerville, I. (2005). *Ingeniería del software*. Pearson Educación.