Plan de Pruebas

“FamCash”

Responsables: Diaz Alvizuri Luis Fabian, Huarcaya Lizarraga Astrid, Pinto Gálvez Sofía Alejandra, Retamozo Vasquez Santiago Sebastian, Rosas Lipa Gianella Ariana, Zeballos Huayna Diego Alonso.

Versión	Fecha	Descripción	Elaboradores
1.0	06/11/2025	Primera versión	Zeballos Huayna Diego Alonso Diaz Alvizuri Luis Fabian Pinto Gálvez Sofía Alejandra Retamozo Vasquez Santiago Sebastian

Cliente:

- Guillermo Enrique Calderón Ruiz

Índice General

1. Introducción

Este documento es el plan de pruebas para el sistema de gestión de economía familiar FamCash , desarrollado por el equipo SOLVE-IT como parte del proyecto de Ingeniería del Software II.

El objetivo principal de este documento es organizar, estructurar y garantizar la correcta verificación y validación del sistema , el cual ha sido propuesto para que un cliente pueda planificar su economía familiar. Esto se logrará mediante la planificación y ejecución de distintos tipos de pruebas.

Este plan se ha elaborado para establecer con claridad los criterios, técnicas, recursos y responsabilidades necesarias para evaluar que el software cumpla con los requisitos funcionales (como el listado de funcionalidades y las nuevas GUIs) y no funcionales definidos. Sirve como una guía detallada para asegurar que los componentes del sistema funcionen correctamente tanto de forma aislada como integrados, reduciendo el riesgo de errores en producción y asegurando una experiencia satisfactoria para el usuario final.

El documento está compuesto por apartados clave como el alcance del plan de pruebas, donde se definen los niveles de prueba a utilizar y sus exclusiones; un cronograma detallado de actividades; una estimación de costos asociados; los recursos humanos (el grupo de 6 estudiantes) y físicos involucrados; y finalmente, el diseño de los casos de prueba.

En conjunto, este plan de pruebas busca ser un instrumento de control y mejora continua durante el proceso de verificación y validación del sistema, facilitando la toma de decisiones, el aseguramiento de la calidad y el cumplimiento de los objetivos del proyecto.

2. Alcance

Esta sección define el alcance del Plan de Pruebas del sistema FamCash, estableciendo los límites y objetivos de las actividades de verificación y validación. El propósito principal es asegurar que los componentes desarrollados del sistema funcionen correctamente, cumplan con los requisitos especificados y estén listos para su despliegue.

Se especifican los tipos de pruebas a realizar, los niveles en los que serán aplicadas, así como las funcionalidades incluidas y aquellas que quedarán fuera del proceso de prueba. Este enfoque permite enfocar los esfuerzos del equipo de aseguramiento de calidad en los elementos críticos del sistema, garantizando una cobertura adecuada y eficiente de los escenarios más relevantes desde la perspectiva técnica y del usuario final.

2.1 Alcance del Plan de Prueba

2.1.1 Tipos de Pruebas

- **Pruebas Unitarias**
  Prueba individual de cada función o método desarrollado por el equipo SOLVE-IT.
  Ejemplo:
  - Función cargarInformación(in colecciónTransacciones:string): void de UI-06-VistaBalance.

- **Pruebas de Integración**
  Verificar la interacción entre los diferentes módulos del sistema FamCash.
  Ejemplo:
  - Integración del módulo UI-05-VistaEntradaDiaria con el módulo UI-06-VistaBalance, asegurando que los ingresos y egresos registrados se reflejen correctamente en el balance.

- **Pruebas de Validación**
  Demostración al cliente (Guillermo Calderón) de que el sistema FamCash cumple con los requisitos y atiende la necesidad de "planificar su economía familiar".
  Metodología:
  - Sesiones guiadas usando los mockups aprobados.
  - Checklist basado en el documento de requisitos(Rastreo de requisitos).

- **Pruebas de Sistema**
  Verificar el comportamiento completo de FamCash en un entorno similar al real (despliegue web).
  Enfoque:
    - Pruebas de usabilidad (críticas para el usuario que "no sabe nada de computación").
    - Compatibilidad en los navegadores web más comunes (Chrome, Firefox).
    - Rendimiento con un número pequeño de usuarios (ej. 5-10 usuarios simultáneos).

2.1.2 Niveles de Prueba

- **Pruebas Alfa**
    - Realizadas por: El propio equipo de desarrollo y testing (SOLVE-IT).
    - Ambiente: Entorno de desarrollo local o de integración.
    - Cobertura: 100% de las funciones críticas (Registro, Entrada Diaria, Balance, Configuración).

- **Pruebas Beta**
    - Participantes: Un grupo selecto de usuarios finales (puede ser el cliente/profesor u otros compañeros que simulan ser el usuario no técnico).
    - Casos: Flujos principales de uso (ej. "registrar un gasto diario", "consultar el balance mensual").

2.1.3 Exclusiones

Para enfocar los esfuerzos del equipo, las siguientes pruebas quedan fuera del alcance de este plan:

- No se probará en hardware o software obsoleto (ej. navegadores fuera de las últimas dos versiones principales).
- No se realizarán pruebas de estrés o carga con un volumen masivo de usuarios (ej. más de 20 usuarios simultáneos).
- No se incluyen pruebas de seguridad avanzadas (como penetration testing).

2.2 Gestión de Alcance

La gestión de cambios en el alcance de pruebas es fundamental para asegurar que las actividades de verificación y validación del sistema FamCash se desarrollen según lo planeado. Se realizarán revisiones periódicas del alcance definido, evaluando su cumplimiento y las posibles desviaciones. El proceso de gestión será el siguiente:

- **Monitoreo y Detección de Cambios:** El equipo SOLVE-IT llevará a cabo revisiones semanales del alcance, detectando adiciones o modificaciones en los requisitos a validar (ej. cambios en las GUIs o en el rastreo de requisitos).
- **Comparación y Evaluación del Cambio:** Se contrastarán los requisitos originales con los propuestos. Si se detectan discrepancias, se documentarán las causas (ej. nuevo requerimiento del cliente , cambio técnico).
- **Acciones Correctivas:** Se tomarán medidas como reasignación de tareas dentro del equipo o ajustes en los casos de prueba para adaptarse a los cambios.
- **Acciones Preventivas:** Se reforzarán prácticas como la revisión temprana de los mockups con el cliente para reducir futuros impactos.
- **Actualización del Alcance:** Cualquier modificación deberá ser aprobada por todo el equipo SOLVE-IT y el cliente (profesor). Una vez aprobada, se reflejará el cambio en la documentación (como el Rastreo de Requisitos FamCash y se notificará a todos los involucrados.

3. Requisitos

Esta sección detalla cómo se verificará el cumplimiento de los requisitos del sistema FamCash a través de actividades de prueba. Los requisitos considerados incluyen aspectos funcionales, no funcionales y de calidad, todos documentados en la etapa previa del proyecto. Las pruebas se han diseñado para cubrir estos requisitos mediante criterios de aceptación claros y una matriz de trazabilidad que permita garantizar una cobertura completa.

3.1 Enfoque Basado en Requisitos

Las pruebas se han diseñado considerando los requisitos del proyecto, los cuales se validarán para asegurar que:

- Las funcionalidades del sistema FamCash se comporten como se espera.
- Las condiciones de uso (principalmente usabilidad y compatibilidad) se cumplan en los entornos previstos.
- El producto entregado al cliente cumpla con los criterios de calidad establecidos

3.2 Tipos de Requisitos Considerados

Los tipos de requisitos considerados para FamCash se muestran en la tabla 3.1.

Tabla 1 Clasificación de requisitos y su estrategia de prueba

Tipo de requisito	Descripción	Estrategia de prueba
Funcionales (RF)	Especifican las funciones que el sistema FamCash debe realizar. Ejemplos: • Permitir el registro de una cuenta familiar (UI-02).	Se verificarán mediante pruebas funcionales y de validación para asegurar que cada función opera según lo especificado en los mockups .
No funcionales (RNF)	Relacionados con las cualidades del sistema. Ejemplos: Usabilidad (Crítico): El sistema debe ser extremadamente intuitivo, ya que el usuario "no sabe nada de computación". Compatibilidad: El sistema debe operar en los navegadores web modernos (Chrome, Firefox).	Se verificarán mediante pruebas de usabilidad (con usuarios que simulen ser el cliente) y pruebas de compatibilidad en distintos navegadores.
De Interfaz (RI) / Calidad	Criterios para evaluar si la interfaz es aceptable. Ejemplo: Fidelidad Visual: La implementación debe ser similar al modelo de navegabilidad.	Verificación visual contra los mockups aprobados para asegurar que la experiencia de usuario es la diseñada.

3.3 Criterios de Aceptación

Cada requisito será considerado cumplido si:

- El caso de prueba asociado ha sido ejecutado con resultado "Exitoso".
- El comportamiento observado en FamCash cumple con los flujos definidos en los mockups (ej. UI-05-VistaEntradaDiaria actualiza correctamente UI-06-VistaBalance).
- Se han documentado las evidencias correspondientes (capturas de pantalla, videos).

3.4 Trazabilidad de Requisitos

Para garantizar que todos los requisitos sean adecuadamente cubiertos, se emplea la Matriz de Trazabilidad de Requisitos (documentada en Rastreo de Requisitos FamCash). Esta matriz vincula los requisitos funcionales y no funcionales con los casos de prueba diseñados.

3.4.1 Matriz de Trazabilidad de Requisitos

La Tabla 2 contiene la Matriz de Trazabilidad, donde cada requisito tiene un identificador, su descripción, el tipo de requisito, los casos de prueba asociados, los criterios de aceptación y su prioridad.

Tabla 2.1 Matriz de trazabilidad de requisitos para el plan de pruebas del sistema FamCash

ID	Descripción del Requisito	Tipo	Casos de Prueba	Criterios de Aceptación	Prioridad
RF-01	Registrar Usuario (con validación de unicidad y contraseña)	Funcional	PU-001	Usuario registrado con éxito. Unicidad de username validada.	Alta
RF-02	Iniciar Sesión (con redirección a selección de perfiles)	Funcional	PU-002	Usuario autenticado y redirigido a la selección de perfiles.	Alta
RF-03	Selección de Perfil Familiar (con verificación de contraseña de perfil)	Funcional	PU-003	Contraseña de perfil validada. Contexto de operación (estándar/admin) establecido.	Alta
RF-04	Registrar Ingreso Diario (monto, concepto, fecha)	Funcional	PU-004	Monto numérico validado. Balance actualizado en tiempo real.	Alta
RF-05	Registrar Egreso Diario (monto, concepto, fecha)	Funcional	PU-004	Monto numérico validado. Balance actualizado en tiempo real.	Alta
RF-06	Guardar Transacciones (con botón de guardado)	Funcional	PI-001	Datos validados y guardados. Confirmación visual mostrada.	Alta
RF-07	Ver Balance Personal (con filtros de período)	Funcional	PV-001	Datos de balance (ingresos, egresos, neto) correctos. Filtros de período funcionales.	Alta

Tabla 2.2 Matriz de trazabilidad de requisitos para el plan de pruebas del sistema FamCash

ID	Descripción del Requisito	Tipo	Casos de Prueba	Criterios de Aceptación	Prioridad
RF-08	Ver Balance Familiar (vista de administrador consolidada)	Funcional	PI-002	Datos consolidados correctos. Vista por persona funcional.	Alta
RF-09	Filtrar por Fechas (seleccionar fecha final en calendario)	Funcional	PV-002	El balance y la lista de transacciones se actualizan según la fecha seleccionada.	Alta
RF-10	Crear Conceptos (validando nombre único)	Funcional	PU-005	Validación de nombre único funciona. Concepto guardado.	Alta
RF-11	Editar Conceptos (modificar nombre, tipo, etc.)	Funcional	PU-006	Los cambios se guardan y se reflejan en futuros registros.	Alta
RF-12	Editar Monto del concepto (en una fecha específica)	Funcional	PU-007,	El monto se actualiza, el balance personal/familiar se recalcula.	Alta
RF-13	Configurar Perfil Personal (actualizar nombre y contraseña)	Funcional	PV-003	Cambio de contraseña exitoso solo si la contraseña actual es correcta.	Media
RF-14	Crear Perfil Familiar (admin: con validación de contraseñas)	Funcional	PV-004	Perfil creado con éxito. Contraseñas coincidentes validadas.	Media
RF-15	Editar Perfil Familiar (admin: dar/quitar permisos)	Funcional	PV-005	Botones "Dar"/"Quitar" se activan/desactivan dinámicamente.	Media
RF-16	Eliminar Perfil Familiar (admin: con confirmación doble)	Funcional	PV-006	Perfil eliminado lógicamente. Transacciones archivadas.	Media
RF-17	Validación de Entrada (mensajes de error específicos)	Funcional	PV-007	Impide el guardado si los datos no son válidos (ej. monto no numérico).	Alta

Tabla 2.3 Matriz de trazabilidad de requisitos para el plan de pruebas del sistema FamCash

ID	Descripción del Requisito	Tipo	Casos de Prueba	Criterios de Aceptación	Prioridad
RF-18	Cerrar Sesión (botón seguro que redirige a login)	Funcional	PV-008	Sesión terminada y credenciales temporales eliminadas.	Alta
RF-19	Mensajes de Retroalimentación (confirmación/error)	Funcional	PV-009	Mensajes claros mostrados para operaciones exitosas o fallidas.	Media
RNF-01	Seguridad (Control de acceso por roles admin/común)	No funcional	PS-001	Acceso a funciones de administrador (RF-08, RF-16) bloqueado para rol "usuario común".	Alta
RNF-02	Usabilidad (Diseño consistente, facilidad < 2 min)	No funcional	PS-002	Flujo intuitivo validado. Tareas básicas completadas en el tiempo estipulado.	Alta
RNF-03	Portabilidad (Funcionar en Chrome, Firefox, Edge, Opera, Safari)	No funcional	PS-003	Sistema funcional y visualmente coherente en todos los navegadores listados.	Alta
RNF-04	Mantenibilidad (Comentarios en código, control de versiones)	No funcional	N/A (Auditoría)	Se valida mediante revisión de código (Code Review), no con un caso de prueba.	Alta
RNF-05	Escalabilidad (Manejar incremento de 5 a 20 usuarios)	No funcional	PS-004	Sistema mantiene rendimiento sin degradación significativa con 20 usuarios concurrentes.	Alta

3.5. Gestión de Requisitos

La gestión de cambios en los requisitos es esencial para preservar la validez del Plan de Pruebas del sistema FamCash. Cualquier modificación en los requisitos funcionales (ej. un cambio en mockup.drawio) o no funcionales puede afectar la cobertura de los casos de prueba y el cronograma. Se establece el siguiente flujo de control:

- **Solicitud formal de cambio:** Toda propuesta de ajuste a un requisito debe presentarse por escrito al equipo SOLVE-IT.
- **Evaluación del impacto:** El equipo de pruebas analiza la solicitud y determina los casos de prueba afectados y las repercusiones en el cronograma.
- **Aprobación:** El cambio se aprueba por consenso del equipo y se notifica al cliente/profesor.
- **Actualización de la documentación:** Una vez aprobado, se refleja el cambio en todos los documentos relevantes (principalmente Rastreo de Requisitos FamCash y este Plan de Pruebas).

La Tabla 3 muestra el formato que se usará para registrar y hacer seguimiento a cada solicitud de cambio de requisitos.

Tabla 3 Ejemplo de solicitud de cambio de requisitos

Campo	Descripción
Código	SCR-001
Fecha de solicitud	24/10/2025
Solicitado por	Zeballos Huayna Diego Alonso (Project Manager)
Requisito(s) afectado(s)	RF-12 (Editar Monto del concepto)
Descripción del cambio	Añadir una funcionalidad para **"eliminar"** un registro de ingreso/egreso en la pantalla de Entrada Diaria, además de la función existente de "editar monto".
Justificación	Un usuario puede registrar un monto por error y necesita eliminar la transacción completa, no solo modificar el valor a cero.
Impacto en pruebas	• Crear nuevo caso de prueba unitario (PU-005) para el método eliminarTransaccion(). • Modificar casos de prueba de validación (PV-004 y PV-005) para incluir el flujo de eliminación. • Incremento estimado de 4 horas de esfuerzo de prueba.
Estado	Pendiente
Fecha de resolución	—

4. Cronograma

Esta sección se enfocará en definir, secuenciar, estimar, desarrollar y controlar el tiempo necesario para ejecutar las pruebas del proyecto FamCash de manera eficiente y dentro de los plazos establecidos. Se definen las actividades relacionadas con los diferentes tipos de pruebas, incluyendo pruebas unitarias, de integración, de validación y de sistema, descomponiéndolas en tareas específicas y manejables. Se tiene en cuenta la secuenciación de las pruebas, identificando dependencias entre ellas y considerando los entregables parciales del desarrollo. Asimismo, se realiza la estimación del tiempo requerido para cada tipo de prueba en función del avance del proyecto y de los recursos asignados. Finalmente, se construye el cronograma de pruebas, analizando su duración, orden de ejecución y criterios de finalización, con el objetivo de garantizar la calidad y funcionamiento correcto del sistema.

4.1 Actividades

Las actividades de prueba se definen en la Tabla 4. Esta contiene el código de la actividad, la tarea específica, las fechas estimadas de inicio y fin, y los responsables (el equipo SOLVE-IT).

Tabla 4.1 Cronograma de actividades para las pruebas del sistema FamCash

Código	Actividad	Tarea	Inicio	Final	Recursos	Costo (S/)
AP-01-01	Diseño de casos de prueba	Diseño de pruebas unitarias	19/10/25	20/10/25	Diaz Alvizuri Luis (Tech Lead), Pinto Gálvez Sofía (Tester)	1,000.00
AP-01-02	Diseño de casos de prueba	Diseño de pruebas de integración	19/10/25	20/10/25	Retamozo Vasquez Santiago (Tech Lead), Huarcaya Lizarraga Astrid (Tester)	1,000.00
AP-01-03	Diseño de casos de prueba	Diseño de pruebas orientadas a fallos	19/10/25	20/10/25	Rosas Lipa Gianella (Tester), Zeballos Huayna Diego (Tester)	1,000.00
AP-02-01	Pruebas unitarias	Construcció n de pruebas unitarias	21/10/25	21/10/25	Diaz Alvizuri Luis (Tech Lead), Pinto Gálvez Sofía (Tester)	1,500.00

Tabla 4.2 Cronograma de actividades para las pruebas del sistema FamCash

Código	Actividad	Tarea	Inicio	Final	Recursos	Costo (S/)
AP-02-02	Pruebas unitarias	Ejecución de pruebas unitarias	22/10/25	22/10/25	Diaz Alvizuri Luis (Tech Lead), Pinto Gálvez Sofía (Tester)	2,293.00
AP-03-01	Pruebas de integración	Construcción de pruebas de integración	23/10/25	23/10/25	Retamozo Vasquez Santiago (Tech Lead), Huarcaya Lizarraga Astrid (Tester)	1,500.00
AP-03-02	Pruebas de integración	Ejecución de pruebas de integración	24/10/25	25/10/25	Retamozo Vasquez Santiago (Tech Lead), Huarcaya Lizarraga Astrid (Tester)	3,293.00
AP-04-01	Pruebas orientadas a fallos	Construcción de pruebas de sistema y fallos	26/10/25	28/10/25	Rosas Lipa Gianella (Tester), Zeballos Huayna Diego (Tester)	1,500.00
AP-04-02	Pruebas orientadas a fallos	Ejecución de pruebas de sistema y fallos	29/10/25	31/10/25	Rosas Lipa Gianella (Tester), Zeballos Huayna Diego (Tester)	3,293.00
				Total		**14,379.00**

4.2 Gestión de Cronograma

La gestión del cronograma de pruebas es fundamental para asegurar que las actividades de verificación y validación del sistema FamCash se desarrollen de acuerdo al calendario establecido, permitiendo identificar a tiempo desviaciones que puedan afectar la calidad o el avance del software.

Se realizarán revisiones periódicas (reuniones diarias cortas) del progreso de las pruebas programadas, evaluando el cumplimiento de los tiempos, la disponibilidad de los miembros del equipo SOLVE-IT y los resultados obtenidos.

- **Monitoreo y Detección de Cambios en el Cronograma:** Se llevarán a cabo revisiones diarias del avance de las pruebas en cada módulo (ej. Módulo de Registro, Módulo de Balance), detectando retrasos o bloqueos.
- **Comparación y Evaluación del Cambio:** Se contrastan los resultados esperados y reales de ejecución. Si se detectan reprogramaciones necesarias, se documentan las causas (por ejemplo, errores bloqueantes en el software, problemas de entorno, indisponibilidad de un miembro del equipo) usando el formato de la Tabla 4.3.
- **Acciones Correctivas:** Se tomarán medidas como reasignación de pruebas entre miembros del equipo o ajustes en el orden de ejecución para mitigar retrasos.
- **Acciones Preventivas:** Se reforzarán prácticas como realizar pruebas paralelas en módulos independientes (ej. probar UI-02 Registro mientras se prueba UI-10 Configuración de Conceptos) para reducir futuros impactos.
- A**ctualización del Cronograma de Pruebas:** Cualquier modificación deberá ser aprobada por todo el equipo SOLVE-IT. Una vez aprobada, se reflejará el cambio en el cronograma.

La Tabla 5 muestra el formato que se utilizará para documentar las desviaciones detectadas durante la ejecución del plan de pruebas.

Tabla 5 Ejemplo de registro de desviaciones detectadas durante la ejecución del plan de pruebas de FamCash

Código	Fecha	Revisor (es)	Actividad Afectada	Causa de la Desviación	Acciones propuestas	Estado	Fecha de resolución
SCC-01	25/10/2025	Santiago Retamozo, Astrid Huarcaya	AP-03-02 (Ejecución pruebas de integración)	Bloqueo en la integración del RF-08 (Ver Balance Familiar) y RF-16 (Crear Perfil Familiar). El endpoint de consolidación de datos no responde.	Pausar la ejecución de AP-03-02. Notificar al equipo de Back-End (Luis Diaz, Sofía Pinto) para corrección prioritaria. Extender la ejecución 1 día.	En evaluación	

5. Costos

Este capítulo detalla los recursos necesarios y la estimación de costos asociados a la ejecución de las actividades de prueba para el sistema FamCash. Dado que se trata de un proyecto académico desarrollado por el equipo SOLVE-IT, el costo principal no es monetario, sino que se mide en Horas-Hombre (HH).

5.1 Metodología de Costeo

Este presupuesto se ha elaborado mediante un proceso estructurado en tres fases:

A. Cálculo individual: Determinación de costos unitarios por rol y recurso
B. Generalización: Proyección escalada para todo el equipo de trabajo
C. Asignación temporal: Distribución según el cronograma de actividades

5.2 Detalle Individual de Costos

5.2.1 Recursos Humanos (Costos Unitarios)

Esta sección detalla la inversión requerida por cada miembro del equipo durante las 5 semanas de ejecución.

Tabla 6 Costo individual por rol

Rol	Horas/Día	Días	Tarifa(S/)	Total
Tester Senior	6	24	35	5,040.00
Tester Junior	8	24	20	3,840.00

Notas

- Días laborales calculados: 24 días (4 semanas × 6 días hábiles)
- Diferenciación horaria basada en complejidad de tareas asignadas
- Tarifas alineadas con el mercado local para perfiles QA

5.2.2 Infraestructura (Costos Unitarios)

Desglose de plataformas tecnológicas requeridas para la ejecución de pruebas.

Tabla 7 Costos unitarios de plataformas

Recurso	Detalle Tecnico	Costo(S/)	Justificación
Heroku Dyno	Plan Professional - 2.5GB RAM	182.50	Entorno estable para pruebas E2E
Selenium Grid	BrowserStack Basic Plan	182.50	Soporte para automatización cross-browser
JMeter Cloud	20 horas de ejecución	548.00	Paquete básico para pruebas de carga

5.3 Generalización para el Equipo

5.3.1 Personal (Equipo Completo)

Proyección escalada para los 5 miembros del equipo de QA.

Tabla 8 Extrapolación de costos de personal

Rol	Cantidad	Costo Unitario	Total	Cantidad
Tester Senior	3	5,040.00	15120	Tester Senior
Tester Junior	3	3,840.00	11520	Tester Junior
Total	6	-	26640	Total

5.3.2 Infraestructura (Consolidado)

Inversión total en plataformas tecnológicas compartidas.

Tabla 9 Costos consolidados de plataformas

Concepto	Total (S/)	Distribución Temporal
Heroku (2 Dynos)	365.00	Semanas 3-4
Selenium Grid	182.50	Semanas 3-4
JMeter Cloud	548.00	Semanas 4
Total	1,095.50	-

5.4 Presupuesto Consolidado

Visión global de la inversión requerida para el plan de pruebas.

Tabla 10 Resumen Final del Presupuesto

Concepto	Monto (S/)	Observaciones
Recursos Humanos	26,640.00	3 seniors + 3 juniors
Infraestructura	1,095.50	Heroku + Selenium + JMeter
Subtotal	27,735.5	-

5.5 Gestión de costos

La gestión de cambios en los costos es fundamental para asegurar que las actividades de verificación y validación del sistema FamCash se desarrollen de acuerdo con el presupuesto establecido. Este control permite identificar tempranamente desviaciones que puedan comprometer la calidad, los plazos o la sostenibilidad financiera del proyecto.

Se realizarán revisiones periódicas de las actividades de prueba programadas, verificando si el uso de recursos humanos y tecnológicos se mantiene dentro de los márgenes previstos. Cualquier diferencia deberá ser documentada, analizada y, si corresponde, corregida mediante ajustes presupuestarios justificados.

- **Monitoreo y Detección de Cambios en los Costos:** Se llevarán a cabo revisiones continuas del uso de recursos durante cada fase de pruebas (unitarias, integración, validación y sistema). Estas evaluaciones buscan detectar incrementos imprevistos en las horas hombre, licencias de software o necesidades de infraestructura adicionales.
- **Comparación y Evaluación del Cambio:** Se contrastan los costos reales ejecutados frente a los estimados. Las discrepancias deben ser analizadas considerando factores como cambios en el alcance, problemas técnicos, disponibilidad de testers, entre otros.
- **Acciones Correctivas:** Se podrán tomar medidas como redistribución de tareas, sustitución de herramientas o modificación en el orden de ejecución, para evitar que se generen sobrecostos en las etapas subsiguientes.
- **Acciones Preventivas:** Se reforzarán prácticas como la planificación cruzada de tareas entre miembros del equipo para evitar desviaciones futuras.
- **Actualización del Presupuesto:** Toda solicitud de modificación en los costos deberá contar con la aprobación del equipo. Una vez aceptada, el nuevo presupuesto será actualizado formalmente y comunicado a todos los involucrados.

6. Recursos

Esta sección detalla los recursos necesarios para la ejecución del plan de pruebas del sistema FamCash. Se garantiza que tanto el equipo humano como los recursos tecnológicos y materiales estén disponibles y sean utilizados de manera óptima durante la verificación y validación del sistema. Para llevar a cabo las actividades de prueba de FamCash, se asegura que se cuente con el personal adecuado, así como con el equipamiento y las herramientas tecnológicas necesarias para la ejecución.

6.1 Recursos Humanos

Para la ejecución del plan de pruebas del sistema FamCash, se asignaron roles específicos a los miembros del equipo con el objetivo de asegurar la cobertura efectiva de todas las actividades planificadas. La tabla 6.1 detalla los integrantes del equipo, el rol asumido y una breve descripción de sus responsabilidades:

Tabla 11 Asignación de roles según el plan de pruebas del sistema Famcash

Nombre	Rol en las pruebas	Responsabilidades principales
Diaz Alvizuri Luis Fabian	Tester Senior / Coordinador de pruebas	Diseñar los casos de prueba, coordinar actividades del equipo, controlar el cronograma, validar entregables, ejecutar pruebas de integración, validación.
Huarcaya Lizarraga Astrid Judith	Tester Senior	Ejecutar y supervisar pruebas unitarias, pruebas de integración, y apoyar en pruebas de validación y sistema.
Pinto Gálvez Sofía Alejandra	Tester Senior	Ejecutar y supervisar pruebas unitarias, pruebas de integración, y apoyar en pruebas de validación y sistema.
Retamozo Vasquez Santiago Sebastian	Tester Junior	Apoyar en pruebas unitarias, participar en pruebas de validación y sistema, documentar resultados
Rosas Lipa Gianella Ariana	Tester Junior	Apoyar en pruebas unitarias, participar en pruebas de validación y sistema, documentar resultados
Zeballos Huayna Diego Alonso	Tester Junior	Apoyar en pruebas unitarias, participar en pruebas de validación y sistema, documentar resultados, apoyar en diseño de casos de prueba.

6.2 Recursos Físicos

- **Equipos informáticos:** Cinco computadoras de escritorio o portátiles con características suficientes para soportar la ejecución de pruebas funcionales, de integración, validación y sistema:
    - Procesador: Intel Core i5 o AMD Ryzen 5.
    - Memoria RAM: Mínimo 8 GB.
    - Almacenamiento: Disco SSD de al menos 256 GB para ejecución ágil de entornos de pruebas y herramientas de testing.
    - Tarjeta Gráfica: Integrada o dedicada, suficiente para pruebas de interfaz gráfica y ejecución de simulaciones ligeras.
    - Sistema Operativo: Windows 10/11, macOS o una distribución Linux compatible.

- **Infraestructura Tecnológica:**
    - Plataformas de prueba: Heroku Dyno para entornos de prueba E2E, Selenium Grid (BrowserStack) para pruebas automatizadas cross-browser, y JMeter Cloud para pruebas de carga.
    - Control de versiones: GitHub (versión Free), para el control y trazabilidad del avance en scripts y configuraciones de prueba.
    - Editor de texto/IDE: Visual Studio Code versión 2025.1 para revisar logs, configurar scripts de prueba automatizados y analizar resultados.
    - Herramientas complementarias: Google Docs y Overleaf para registrar hallazgos e incidencias; draw.io y StarUML para validar consistencia visual y funcional en interfaces.
    - Base de Datos: MySQL-XAMPP (versión 17.4) para pruebas relacionadas con integridad de datos y validaciones de back-end.

6.3 Gestión de Recursos

La gestión de recursos en el plan de pruebas del proyecto FamCash es esencial para asegurar que las actividades de validación, verificación y ejecución de pruebas cuenten con los recursos humanos y tecnológicos necesarios en el momento adecuado. Ante cualquier solicitud de modificación en los recursos, se activa un proceso estructurado que inicia con la evaluación del impacto del cambio sobre las asignaciones de pruebas, y culmina con la actualización de los documentos correspondientes. Este proceso busca garantizar que las decisiones estén alineadas con los objetivos de calidad del software y no afecten negativamente los tiempos del cronograma de pruebas. Solicitud Formal de Cambio: Toda solicitud de modificación al plan de recursos de pruebas debe ser presentada de forma formal usando el formato de la tabla 12. Esta debe incluir una descripción del cambio propuesto, su justificación técnica o estratégica, y una estimación del impacto que tendría en el cronograma y los costos del proceso de pruebas. Evaluación de la Solicitud de Cambio: Se analiza el impacto de la solicitud en la asignación de testers, la disponibilidad de plataformas, herramientas y su efecto sobre el cronograma de ejecución de pruebas. Se evalúan riesgos como retrasos, sobrecarga de tareas o conflictos de planificación. Para ello, se utiliza el cronograma de pruebas y el diagrama de Gantt como herramientas de análisis. Aprobación y Actualización del Cambio: Si se justifica el cambio, este debe ser aprobado por el equipo completo.

Posteriormente, se actualizan los documentos del plan de pruebas, asignaciones de roles y cronograma correspondiente. De ser necesario, también se redistribuyen responsabilidades o se modifican los entornos y herramientas requeridas. Actualización del Cronograma y Registro de Cambios: Cualquier modificación aprobada se refleja en el cronograma de pruebas, detallando los nuevos recursos asignados y fechas reprogramadas. Estos cambios serán comunicados a todo el equipo y se documentarán formalmente para su trazabilidad. La Tabla 6.2 muestra un ejemplo completo de una solicitud de cambio aplicada durante el proceso de pruebas del sistema FamCash. Este tipo de formato permite documentar de manera clara y estructurada cualquier modificación que pueda impactar en los recursos del proyecto.

Tabla 12 Ejemplo de solicitud de cambio de recursos

Campo	Descripción
Código	SCR-001
Fecha de solicitud	30/10//2025
Solicitado por	Diego Zeballos
Descripción del cambio	Se solicita reasignar a Sofía Pinto (Tester Junior) desde la tarea AP-04-02 (Ejecución pruebas de validación) hacia AP 03-02 (Ejecución pruebas de integración), debido a mayor carga de pruebas en esta última actividad.
Justificación técnica o estratégica	Durante la ejecución de AP-03-02 se identificaron más escenarios de integración de lo previsto, lo cual exige mayor esfuerzo en ejecución y documentación.
Impacto en cronograma	Se anticipa reducción de 1 día en la finalización de AP-03 02. No se afecta el inicio de AP-04-02.
Impacto en costos	No se genera costo adicional, se mantiene la misma distribución de horas hombre.
Estado del cambio	Aprobado
Fecha de resolución	31/10/2025

7. Diseño de Casos de Prueba

Esta sección presenta el diseño de los casos de prueba desarrollados para verificar y validar las funcionalidades del sistema FamCash. El diseño de pruebas tiene como objetivo asegurar que cada componente del sistema cumple con los requerimientos establecidos, anticipando posibles fallos y garantizando un comportamiento esperado bajo diferentes condiciones. Para ello, se han diseñado pruebas unitarias, de integración, de validación y de sistema, cubriendo así distintos niveles de prueba.

7.1 Diseño de Pruebas Unitarias

Las pruebas unitarias validan el funcionamiento de componentes individuales del sistema. Los datos de prueba son genéricos y cumplen con las reglas de negocio definidas.

Tabla 13.1 Diseño de caso de prueba unitario PU-001: Verificación lógica construcción

Código	PU-001
Técnica de prueba	Caja blanca.
Elementos de software	Lógica de construcción de la consulta $query para obtener perfiles (Líneas 23-40 de seleccionperfil.php).
Descripción de la prueba	El objetivo es asegurar que la consulta de la base de datos para obtener los perfiles a mostrar en la interfaz (UI-04 - Selección de Perfiles) sea la correcta y compatible. Se verifica que la consulta SQL filtre por el estado 'Habilitado' si la columna estado existe. Si la columna no existe, la consulta debe traer todos los perfiles para mantener la compatibilidad del sistema.
Necesidades del entorno (Precondiciones)	Un entorno de pruebas PHP con una conexión simulada o real a una base de datos MySQL/MariaDB con una tabla Perfil. Necesidad específica: La capacidad de simular la presencia o ausencia de la columna estado en la tabla Perfil durante la ejecución de la prueba.
Especificación de entradas (Datos de prueba)	• $_SESSION["id_familia"] (ID de Familia Autenticada): Un número entero válido (ej. 1). • Simulación de la base de datos: Se simula la existencia de la columna estado en la tabla Perfil (para este caso de éxito).
Especificación de salidas (Resultado esperado)	• Consulta SQL: La cadena de texto de la consulta SQL ($query) debe ser construida con o sin la condición AND estado = 'Habilitado', según corresponda. • Array de Perfiles ($perfiles): Se espera un array que contenga solo los objetos (registros) de los perfiles que el usuario puede ver y seleccionar, filtrados por la familia ($idFamilia) y, si aplica, por el estado 'Habilitado'.

Tabla 13.2 Diseño de caso de prueba unitario PU-001: Verificación lógica construcción

Requisitos de procedimiento (Ejecución)	1. Preparación (Setup): Asegurarse de que el entorno de pruebas simule la existencia de la columna estado en la tabla Perfil y poblar la tabla con perfiles (ej. Perfil A - Habilitado, Perfil B - Deshabilitado, ambos de la Familia 1). 2. Ejecución: Iniciar la sesión con $_SESSION["id_familia"] = 1 y ejecutar la sección del código responsable de la construcción de $query y la consulta a la base de datos. 3. Verificación: Confirmar que la variable $query contenga el filtro estado = 'Habilitado'. 4. Verificación de Resultados: Confirmar que el array $perfiles solo contenga el Perfil A (Habilitado).
Dependencia entre casos (Criterio de éxito)	CU-01 Registrar cuenta, CU-10-1 Crear conceptos, CU-11 Crear perfil

Tabla 14.1 Diseño de caso de prueba unitario PU-002: Verificación contraseña y sesión

Código	PU-002
Técnica de prueba	Caja blanca.
Elementos de software	Lógica de verificación de contraseña (password_verify) y asignación de sesión tras solicitud POST (Líneas 44-59 del seleccionperfil.php).
Descripción de la prueba	El objetivo es asegurar la correcta autenticación del perfil seleccionado. Se debe verificar que si la contraseña ingresada en el modal (UI-04) coincide con el hash almacenado, el sistema actualiza las variables de sesión (perfil_id, perfil_nombre, perfil_rol) y prepare la redirección a la Interfaz de Entrada Diaria (UI-05).
Necesidades del entorno (Precondiciones)	Entorno de pruebas PHP que soporte sesiones ($_SESSION) y pueda simular solicitudes HTTP POST. Base de datos poblada con un perfil y su hash de contraseña.
Especificación de entradas (Datos de prueba)	• Simulación de solicitud POST: Se envía $_POST["perfil_id"] (ej. 101) y $_POST["password"] (ej. "miClaveSecreta"). • Base de Datos: El Perfil 101 debe existir y tener un hash de la contraseña "miClaveSecreta"
Especificación de salidas (Resultado esperado)	• Variables de Sesión: $_SESSION["perfil_id"] = 101, $_SESSION["perfil_nombre"] = "Papá", $_SESSION["perfil_rol"] = "Administrador". (Si es exitoso) • Salida (Script): Se espera que el código PHP imprima el script de éxito: redirigirAInterfazDentroDeCuenta(5);

Tabla 14.2 Diseño de caso de prueba unitario PU-002: Verificación contraseña y sesión

Requisitos de procedimiento (Ejecución)	1. Preparación (Setup): Asegurarse de que la tabla Perfil contenga un perfil de prueba (ej. ID 101) con una contraseña hasheada (ej. hash("miClaveSecreta")). 2. Ejecución: Simular una solicitud POST con perfil_id=101 y password="miClaveSecreta" (contraseña correcta). 3. Verificación de Sesión: Verificar que las variables $_SESSION["perfil_id"], $_SESSION["perfil_nombre"], y $_SESSION["perfil_rol"] se hayan establecido con los datos del Perfil 101. 4. Verificación de Salida: Verificar que el script de éxito (redirigirAInterfazDentroDeCuenta(5)) se haya emitido.
Dependencia entre casos (Criterio de éxito)	CU-01 Registrar cuenta (Para obtener id_familia), CU-11-1 Crear perfil (Para que exista el perfil a autenticar).

7.2 Diseño de Pruebas de Integración

Las pruebas de integración validan la interacción entre múltiples componentes del sistema. Se muestran los diseños de integración del sistema FamCash.

Tabla 15.1 Diseño de caso de prueba de integración PI-001: Verificación del método editarConcepto

Código	PI-001
Elemento a evaluar	Funcionalidad de edición de conceptos y su correcta visualización en la interfaz.
Técnica de prueba	Caja Negra.
Descripción de la prueba	Se prueba el correcto desempeño del flujo completo de edición de un concepto previamente registrado, verificando que los cambios se reflejen correctamente.
Necesidades del entorno (Precondiciones)	Un sistema operativo de ventana Windows 10/11 o distribución Linux compatible (basada en kernel 5.4+), servidor de XAMPP con Apache 2.4+ y PHP 7.4+. Un MySql(El mismo que incorpora el XAMPP) con datos poblados respecto a Conceptos, procesador Intel i3, AMD Ryzen 3, equivalente o superior, 8 GB de RAM, 256 GB de disco (preferiblemente SSD) y al menos un concepto previamente registrado en el sistema.
Especificación de entradas (Datos de prueba)	• Cadenas no nulas de caracteres alfanuméricos para el campo nombre. • Selección de tipo: Ingreso o Egreso. • Valor numérico decimal positivo para el campo monto (formato: XX.XX). • Selección de periodo: Diario, Semanal, Quincenal, Mensual, Ninguno. • Fechas válidas seleccionadas mediante calendario para fecha inicio y fecha fin (opcional). • Concepto previamente registrado en el sistema.
Especificación de salidas (Resultado esperado)	• El sistema debe guardar exitosamente los cambios realizados. • Redirección automática a la interfaz visualizar conceptos. • Los cambios persisten en la base de datos MySql.

Tabla 15.2 Diseño de caso de prueba de integración PI-001: Verificación del método editarConcepto

Requisitos de procedimiento (Ejecución)	1. Introducir el nombre o categoría de un concepto previamente registrado en la barra de búsqueda y localizarlo. 2. Hacer clic en el botón 'Editar' del concepto seleccionado (UI-07). 3. Verificar que se carga la interfaz con los datos actuales del concepto. 4. Modificar el campo nombre con un nuevo valor válido. 5. Seleccionar una categoría diferente desde el desplegable (o crear una nueva si es necesario). 6. Cambiar el tipo entre 'Ingreso' y 'Egreso'. 7. Modificar el monto ingresando un valor numérico decimal. 8. Seleccionar un periodo diferente (si se elige 'Personalizado', ingresar el número de días). 9. Modificar las fechas de inicio y fin usando el selector de calendario. 10. Hacer clic en el botón 'Guardar concepto'. 11. Verificar redirección automática a la visualización del concepto (UI-07). 12. Localizar el concepto editado y verificar los cambios en la tabla (usar búsqueda si es necesario).
Dependencia entre casos (Criterio de éxito)	CU-011-CrearConcepto

Tabla 16.1 Diseño de caso de prueba de integración PI-002: Verificación del método seleccionarPerfil

Código	PI-002
Elemento a evaluar	Funcionalidad de Selección de Perfil y validación de la contraseña para ingresar al sistema desde la interfaz de perfiles.
Técnica de prueba	Caja Negra.
Descripción de la prueba	Se prueba el flujo de selección de un perfil familiar y la autenticación exitosa con su contraseña. Se verifica que un perfil solo pueda acceder si proporciona la contraseña correcta.
Necesidades del entorno (Precondiciones)	Un sistema operativo de ventana Windows 10/11 o distribución Linux compatible (basada en kernel 5.4+), servidor de XAMPP con Apache 2.4+ y PHP 7.4+. Un MySql(El mismo que incorpora el XAMPP) con datos poblados respecto a perfiles, procesador Intel i3, AMD Ryzen 3, equivalente o superior, 8 GB de RAM, 256 GB de disco (preferiblemente SSD) y al menos un perfil previamente registrado en el sistema.
Especificación de entradas (Datos de prueba)	• La interfaz UI-04-SelecciónDePerfiles debe estar visible. • Al menos un perfil familiar (ej. Papá) debe estar visible y disponible para selección. • La contraseña asociada al perfil seleccionado debe ser conocida.
Especificación de salidas (Resultado esperado)	• Al ingresar la contraseña correcta, el usuario debe ser redirigido a la interfaz principal (UI-05-VistaEntradaDiaria). • Al ingresar una contraseña incorrecta, el sistema debe mostrar un mensaje de error y mantener al usuario en la pantalla de selección de perfiles.

Tabla 16.2 Diseño de caso de prueba de integración PI-002: Verificación del método seleccionarPerfil

Requisitos de procedimiento (Ejecución)	1. Navegar a la URL de perfiles: https://www.FamCash.com/perfiles (UI-04). 2. Hacer clic en uno de los perfiles mostrados (ej. Papá). 3. Verificar que se muestre el pop-up o sección para ingresar su contraseña. 4. Prueba Positiva (Acceso Exitoso): Ingresar la contraseña correcta en el campo Contraseña. 5. Hacer clic en el botón Aceptar. 6. Verificar la redirección exitosa a la vista de Entrada diaria (UI-05). 7. Prueba Negativa (Contraseña Incorrecta): Repetir los pasos 1 a 3. 8. Ingresar una contraseña incorrecta en el campo Contraseña. 9. Hacer clic en el botón Aceptar. 10. Verificar que se muestre un mensaje de error de autenticación y que el usuario permanezca en la UI-04.
Dependencia entre casos (Criterio de éxito)	CU-001-Registrarse y CU-002-IniciarSesion

7.3 Diseño de Pruebas de Validación

Las pruebas de validación tienen como propósito asegurar que los componentes del sistema FamCash satisfacen los requisitos funcionales, evaluándolos desde la perspectiva del usuario. Se muestran los diseños de validación para FamCash.

Tabla 17 Diseño de caso de prueba de validación PV-001: Verificación formulario de crearUsuario

Código	PV-001
Elemento a evaluar	El correcto funcionamiento del formulario de Creación de Usuario (Registro).
Técnica de prueba	Caja Negra.
Descripción de la prueba	Esta prueba verificará que un nuevo cliente puede registrarse exitosamente en el sistema, proveyendo todos los datos requeridos, y que el sistema valida correctamente las entradas (como nombres de usuario duplicados o contraseñas que no coinciden).
Necesidades del entorno (Precondiciones)	Un sistema operativo (Windows/Linux), servidor XAMPP (Apache 2.4+ y PHP 7.4+), y un servidor PostgreSQL 12+ con la base de datos del proyecto ya creada.
Especificación de entradas (Datos de prueba)	• Campo "Nombre de Usuario": Cadena alfanumérica no nula. • Campo "Nombre": Cadena alfanumérica no nula. • Campo "Rol": Selección válida ("Administrador Familiar" o "Familiar"). • Campo "Contraseña": Cadena no nula de caracteres. • Campo "Confirmar Contraseña": Debe ser una cadena idéntica a la del campo "Contraseña".
Especificación de salidas (Resultado esperado)	• Si el registro es exitoso: El sistema debe mostrar un mensaje de confirmación, persistir el usuario en la BD (PostgreSQL) y redirigir a la página de Inicio de Sesión. • Si el nombre de usuario ya existe: El sistema debe mostrar un mensaje de error claro y no debe crear el usuario. • Si las contraseñas no coinciden: El sistema debe mostrar un mensaje de error y no debe crear el usuario.

Tabla 18 Diseño de caso de prueba de validación PV-002: Verificación formulario de configurarPerfilPersonal

Código	PV-002
Elemento a evaluar	El correcto funcionamiento del formulario para modificar los datos del usuario actualmente logueado.
Técnica de prueba	Caja Negra.
Descripción de la prueba	Esta prueba verificará que el cliente puede actualizar su información personal (como el nombre o el rol) a través de su perfil, y que los cambios se guardan correctamente.
Necesidades del entorno (Precondiciones)	Un sistema operativo (Windows/Linux), servidor XAMPP (Apache+PHP) y PostgreSQL. Un usuario debe estar previamente registrado e iniciar sesión en el sistema para acceder a su perfil.
Especificación de entradas (Datos de prueba)	• Usuario actualmente logueado en el sistema. • Campo "Nombre": Cadena alfanumérica no nula. • Campo "Rol": Selección de "Administrador Familiar" o "Familiar".
Especificación de salidas (Resultado esperado)	• Si la modificación es exitosa: El sistema debe mostrar un mensaje de confirmación, y reflejar los cambios en la interfaz y persistirlos en la BD. • Si se intenta guardar con datos inválidos: debe mostrar un mensaje de error apropiado y no permitir la actualización.

7.4 Diseño de Pruebas de Sistema

Las pruebas de sistema tienen como propósito evaluar el comportamiento del sistema FamCash como una unidad integrada, verificando que sus componentes colaboran correctamente bajo diferentes condiciones y cumplen los requisitos especificados. Estas pruebas abarcan funcionalidades completas, así como características no funcionales como seguridad, compatibilidad y rendimiento. Se muestran los diseños de sistema para FamCash.

Tabla 19.1 Diseño de caso de prueba de sistema PS-001: Verificación del sistema

Código	PS-001
Elemento a evaluar	Todo el sistema
Técnica de prueba	Caja Negra.
Descripción de la prueba	Buscar probar todas las partes críticas del sistema (Registro de Entrada Diaria, gestión de usuarios, conceptos y categorías; visualización del historial de conceptos antes y después de modificación de estos mismos) asegurando su funcionamiento correcto en un sistema operativo Linux.
Necesidades del entorno (Precondiciones)	Un sistema operativo de distribución Linux compatible (basada en kernel 5.4+), servidor de XAMPP con Apache 2.4+ y PHP 7.4+. Un MySql(El mismo que incorpora el XAMPP)l con datos poblados, procesador Intel i3, AMD Ryzen 3, equivalente o superior, 8 GB de RAM, 256 GB de disco (preferiblemente SSD).
Especificación de entradas (Datos de prueba)	• Cadenas no nulas de caracteres alfanuméricos para el nombre y descripción. • Cadenas no nulas de caracteres alfanuméricos para el usuario, nombre y contraseñas para los Usuarios. • Selección de rol: Administrador Familiar o Familiar. • Cadenas no nulas de caracteres alfanuméricos para el campo nombre, categoría. • Selección de tipo: Ingreso o Egreso. • Valor numérico decimal positivo para el campo monto (formato: XX.XX). • Selección de periodo: Diario, Semanal, Quincenal, Mensual, Personalizado o Eventual. • Fechas válidas seleccionadas mediante calendario para fecha inicio y fecha fin (opcional).
Especificación de salidas (Resultado esperado)	Al finalizar, el sistema debe poder haber registrado exitosamente una Entrada Diaria, cargar y editar información ya guardada en la base de datos, en caso de registrar nueva información se guarde y sea persistente en la base de datos MySql además de ser reflejada en la vista correspondiente.

Tabla 19.2 Diseño de caso de prueba de sistema PS-001: Verificación del sistema

Requisitos de procedimiento (Ejecución)	1. Ejecutar y registrarse en el sistema desde el sistema operativo Linux. 2. Añadir nuevos usuarios, conceptos y categorías. 3. Realizar un Registro de Entrada Diaria y verificar su visualización en el Balance. 4. Cargar un concepto previamente registrado y verificar su coherencia con lo registrado. 5. Editar los datos del concepto y confirmar los cambios. 6. Confirmar los cambios reflejados en la visualización de conceptos además de que estos cambios sean reflejados en MySql
Dependencia entre casos (Criterio de éxito)	CU-08 Gestionar usuarios, CU-08-1 Crear usuarios, CU-08-2 Editar usuario, CU-09 Gestionar conceptos, CU-09-1 Crear conceptos, CU-09-2 Editar conceptos, CU-10 Gestionar categorías, CU-10-1 Crear categorías, CU-10-2 Editar categorías.

Tabla 20.1 Diseño de caso de prueba de sistema PS-002: Verificación compatibilidad sistema

Código	PS-002
Elemento a evaluar	Compatibilidad del sistema con el entorno macOS, específicamente la interfaz de usuario (UI) y la persistencia de datos tras la creación y modificación de Perfiles Familiares.
Técnica de prueba	Caja Negra.
Descripción de la prueba	Verificar que todos los elementos visuales de las interfaces de usuario (UI-04, UI-05, etc.) se representen correctamente en macOS y que las funcionalidades de gestión de perfiles (creación, edición, eliminación) se ejecuten sin errores en este entorno.
Necesidades del entorno (Precondiciones)	Un sistema operativo macOS (versión reciente), servidor de MAMP/XAMPP compatible con Apache 2.4+ y PHP 7.4+. Un MySql(El mismo que incorpora el XAMPP) y el sistema con datos de al menos un perfil Administrador.
Especificación de entradas (Datos de prueba)	• Credenciales válidas para un usuario Administrador Familiar. • Datos de entrada válidos para crear un nuevo perfil familiar. • Datos de entrada válidos para editar un perfil existente.
Especificación de salidas (Resultado esperado)	• La interfaz de usuario debe cargarse y funcionar sin problemas de diseño o rendering en macOS. • El Administrador debe poder crear, editar y eliminar perfiles exitosamente. • Los cambios en los perfiles deben ser persistentes en la base de datos MySql.
Requisitos de procedimiento (Ejecución)	1. Ejecutar el sistema en el entorno macOS e Iniciar Sesión como Administrador. 2. Verificar la correcta visualización de la interfaz Entrada Diaria (UI-05) y Balance (UI-06). 3. Navegar a Config. Perfiles Familiares y hacer clic en Crear. 4. Crear un nuevo perfil de usuario (ej. Hijo) y verificar que el perfil aparezca en la lista. 5. Navegar a la pestaña Editar. 6. Seleccionar el perfil recién creado y hacer clic en Eliminar (PI-04). 7. Verificar que el perfil haya sido eliminado de la base de datos y la lista de perfiles. 8. Cerrar Sesión y verificar que la Selección de Perfiles (UI-04) se vea correctamente.
Dependencia entre casos (Criterio de éxito)	CU-002-IniciarSesion y CU-006-GestionarPerfilesFamiliares

8. Referencias

Object Management Group. Unified Modeling Language (UML). https://www.omg.org/spec/UML/