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	30/10/2025	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.

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 12.1 Diseño de caso de prueba unitario PU-005: Verificación del método crearConcepto

Código	PU-005
Nombre del caso	Verificación de la creación de un nuevo concepto de ingreso quincenal.
Técnica de prueba	Caja blanca.
Elementos de software	Componente: gestorConceptos (GES-004) Método bajo prueba: crearConcep(nombre: String, tipo: String, periodo: String, dia: int)
Objetivo	Verificar que el método crearConcepvalida las entradas, comprueba duplicados y registra correctamente un nuevo concepto en la tabla Conceptos (TAB-004).
Necesidades del entorno (Precondiciones)	Software: El componente gestorConceptos (GES-004) debe estar instanciado y operativo. Se requiere acceso de lectura y escritura a la tabla Conceptos (TAB-004) de la base de datos de pruebas.
Especificación de entradas (Datos de prueba)	"Ventas" (Restricción: Cadena de 1 a 50 caracteres). tipo: "Ingreso" (Restricción: Debe ser "Ingreso" o "Egreso"). período: "Quincenal" (Restricción: Debe ser "Ninguno", "Diario", "Semanal", "Quincenal" o "Mensual"). día: 15 (Restricción: Entero entre 1 y 31). Relaciones entre entradas: El parámetro día es obligatorio y se utiliza porque el periodo es "Quincenal".

Tabla 12.2 Diseño de caso de prueba unitario PU-005: Verificación del método crearConcepto

Requisitos de procedimiento (Ejecución)	Se debe invocar el método crearConcep con los datos de prueba. Se debe trazar la ejecución interna para: - Verificar que el método ejecuta primero una consulta SELECT COUNT(*) en Conceptos (TAB-004) con WHERE nombre = "Ventas" para validar que no existan duplicados. - Verificar que la lógica de validación interna para tipo y periodo se ejecuta y pasa. - Verificar que se construye y ejecuta la sentencia INSERT en Conceptos (TAB-004) con los valores de entrada. - Verificar que la transacción de la base de datos recibe la orden de commit().
Especificación de salidas (Resultado esperado)	El método retorna un entero positivo (el nuevo idConcepto generado por la TAB). Mapeo a entradas: Un nuevo registro existe en la tabla Conceptos (TAB-004) exactamente con los campos: nombre="Ventas", tipo="Ingreso", periodo="Quincenal", dia=15. El tiempo de respuesta de la función debe ser inmediato (< 150ms).
Dependencia entre casos (Criterio de éxito)	El método valida correctamente la no duplicidad del nombre y ejecuta el INSERT en la TAB. Dependencia: Este caso PU-003 es un prerrequisito para PU-004 (Editar Concepto) y PU-005 (Eliminar Concepto). Si PU-003 falla, dichos casos deben ser bloqueados en la matriz de trazabilidad.

Tabla 13.1 Diseño de caso de prueba unitario PU-006: Verificación del método editarConcepto

Código	PU-006
Nombre del caso	Verificación de la edición exitosa de un concepto (cambio de nombre y tipo).
Técnica de prueba	Caja blanca.
Elementos de software	Componente: gestorConceptos (GES-004). Método bajo prueba: editarConcepto(idConcepto: int, nuevoNombre: String, nuevoTipo: String, nuevoPeriodo: String, nuevoDia: int)
Objetivo	Verificar que el método editarConcepto actualiza correctamente los atributos de un concepto existente en la tabla Conceptos (TAB-004), valida el nuevo nombre por unicidad y recalcula las transacciones afectadas si el tipo cambia.
Necesidades del entorno (Precondiciones)	Software: El componente gestorConceptos (GES-004) debe estar instanciado y operativo. Se requiere acceso de lectura y escritura a las tablas Conceptos (TAB-004) y Transacciones (TAB-005) de la base de datos de pruebas. Datos: Debe existir previamente un concepto con idConcepto = 101 y valores originales: nombre="Ventas", tipo="Ingreso", periodo="Quincenal", dia=15.
Especificación de entradas (Datos de prueba)	idConcepto: 101 (Debe existir previamente). nuevoNombre: "Ventas y Servicios" (Restricción: Cadena de 1 a 50 caracteres). nuevoTipo: "Egreso" (Restricción: Debe ser "Ingreso" o "Egreso"). nuevoPeriodo: "Mensual" (Restricción: Debe ser "Ninguno", "Diario", "Semanal", "Quincenal" o "Mensual"). nuevoDia: 1 (Restricción: Entero entre 1 y 31).

Tabla 13.2 Diseño de caso de prueba unitario PU-006: Verificación del método editarConcepto

Requisitos de procedimiento (Ejecución)	Se debe invocar el método editarConcepto con los datos de prueba (101, "Ventas y Servicios", "Egreso", "Mensual", 1). Se debe trazar la ejecución interna para: 1. Validación de Unicidad: Verificar que se ejecuta una consulta SELECT COUNT(*) en Conceptos con WHERE nombre = "Ventas y Servicios" AND idConcepto != 101 para validar que el nuevo nombre no esté duplicado. 2. Validación de Tipo: Verificar si el nuevoTipo ("Egreso") es diferente del tipo original ("Ingreso"). Si es diferente, la lógica interna debe activar un proceso de re-cálculo/actualización de los balances afectados por transacciones anteriores ligadas a este concepto. 3. Ejecución de UPDATE: Verificar que se construye y ejecuta la sentencia UPDATE en la tabla Conceptos (TAB-004) con el idConcepto = 101 y los nuevos valores de entrada. 4. Commit: Verificar que la transacción de la base de datos recibe la orden de commit().
Especificación de salidas (Resultado esperado)	El método retorna un valor booleano o entero que indica el éxito de la operación (ej. True o 1). Mapeo a entradas: El registro con idConcepto = 101 en la tabla Conceptos (TAB-004) debe tener exactamente los campos actualizados: nombre="Ventas y Servicios", tipo="Egreso", periodo="Mensual", dia=1. Rendimiento: El tiempo de respuesta de la función debe ser inmediato (< 200ms).
Dependencia entre casos (Criterio de éxito)	El método valida correctamente el nuevo nombre por no duplicidad, maneja el cambio de tipo de concepto si aplica, y ejecuta exitosamente el UPDATE en la TAB.

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 14 Diseño de caso de prueba de integración PI-001: Verificación del método guardarTransacciones

Código	PI-001
Elemento a evaluar	Flujo completo de registro y guardado de una Transacción (Ingreso/Egreso) y la integración entre la Interfaz de Usuario y la Base de Datos.
Técnica de prueba	Caja Negra.
Descripción de la prueba	Se prueba el flujo de usuario para registrar una transacción diaria, verificando que los datos se validen correctamente en la interfaz, se persistan en la tabla Transacciones de la base de datos y que el balance (Balance Personal y Balance Familiar) se actualice en tiempo real (según criterios de aceptación RF-04 y RF-05).
Necesidades del entorno (Precondiciones)	Un entorno de desarrollo/pruebas con el software y hardware especificado (PostgreSQL, XAMPP, etc.). El sistema debe tener al menos un Usuario, un Perfil y un Concepto registrado accesible para la prueba. El componente de Registro de Transacciones y el componente de Cálculo de Balance deben estar integrados y funcionales.
Especificación de entradas (Datos de prueba)	1. Datos de Transacción: Monto (150.75), Concepto ("Salario"), Fecha ("2023-10-27"). 2. Validación: Se debe utilizar un valor numérico válido para el monto (Formato: XX.XX). 3. Usuario/Perfil: Sesión activa con un Perfil válido (el contexto de la transacción). 4. Acción: Presionar el botón de "Guardar" en la interfaz de registro.
Especificación de salidas (Resultado esperado)	1. Persistencia: Un nuevo registro con los datos exactos debe existir en la tabla Transacciones (TAB-005) ligado al idPerfil y idConcepto correspondientes (Ver Diagrama de Clases). 2. Retroalimentación: El sistema debe mostrar un mensaje de confirmación visual ("Transacción guardada con éxito"). 3. Integración/Balance: El cálculo del balance total del Perfil activo (entidad Balance en el Diagrama de Clases) debe reflejar el incremento de $150.75 de manera inmediata.

Tabla 15 Diseño de caso de prueba de integración PI-002: Verificación del método verBalanceFamiliar

Código	PI-002
Elemento a evaluar	Consolidación de datos financieros y la correcta visualización del Balance Familiar (Ingresos, Egresos, Neto) de todos los perfiles en la vista de Administrador.
Técnica de prueba	Caja Negra.
Descripción de la prueba	Se prueba el flujo de acceso a la vista de Balance Familiar por un usuario con rol Administrador, verificando que el sistema agregue correctamente las transacciones de todos los perfiles, incluyendo la funcionalidad de vista por persona (RF-08 criterio).
Necesidades del entorno (Precondiciones)	Se requiere el entorno de pruebas estándar. Datos críticos: La base de datos (PostgreSQL) debe tener al menos dos perfiles familiares registrados, y cada uno debe tener al menos una transacción registrada y consolidada, garantizando que el rol del usuario actual sea Administrador.
Especificación de entradas (Datos de prueba)	1. Usuario/Perfil: Inicio de sesión con un perfil que tenga rol de Administrador (RNF-01, PS-001). 2. Datos de Base: La base de datos debe contener transacciones para múltiples perfiles (ej. "Perfil A" con Ingresos/Egresos y "Perfil B" con sólo Egresos). 3. Acción: Navegar a la sección de Balance Familiar (Vista Consolidada) e intentar usar el filtro de "Vista por persona".
Especificación de salidas (Resultado esperado)	1. Consolidación Correcta: El balance mostrado debe ser la suma total de los ingresos y egresos de todos los perfiles familiares activos. 2. Integración: La consulta ejecutada debe interactuar con las tablas Transacciones y Perfil para realizar la agregación de datos (según el Diagrama de Clases). 3. Filtro por Persona: La opción para ver datos por persona (RF-08 criterio) debe funcionar, mostrando el balance exclusivo del perfil seleccionado, y el balance consolidado debe reflejarse en la interfaz. 4. Rendimiento: La carga del balance consolidado (que involucra datos de todos los perfiles) debe ser rápida (< 500ms).

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 16 Diseño de caso de prueba de validación PV-002: Verificación del método filtrarPorFechas

Código	PV-002
Elemento a evaluar	Funcionalidad del filtrado por fechas en las vistas de Balance Personal (RF-07) y Balance Familiar (RF-08).
Técnica de prueba	Caja Negra.
Descripción de la prueba	Se valida que el usuario puede seleccionar una fecha final en el calendario y que tanto el monto del balance (ingresos, egresos, neto) como la lista detallada de transacciones se actualizan para mostrar *solo* los datos comprendidos en el período seleccionado.
Necesidades del entorno (Precondiciones)	Entorno de pruebas estándar. Se requiere que el sistema tenga datos históricos suficientes (transacciones de diferentes meses) para poder verificar que el filtro excluye correctamente los datos fuera del rango. El componente de calendario (datepicker) debe ser funcional.
Especificación de entradas (Datos de prueba)	1. Usuario/Perfil: Inicio de sesión con un perfil que tenga transacciones registradas en diferentes meses (ej. Octubre y Noviembre). 2. Datos de Base: Transacciones registradas: T1 (Fecha: 01/10/2023), T2 (Fecha: 15/10/2023). 3. Acción 1: Abrir el selector de fecha y establecer la fecha en 15/10/2023 3. Acción 2: Abrir el selector de fecha y establecer la fecha en 01/10/2023
Especificación de salidas (Resultado esperado)	1. Balance Actualizado: El Balance Personal/Familiar (RF-07, RF-08) debe mostrar un cálculo que sólo incluya las transacciones T1 y T2. 2. Lista Actualizada: La lista de transacciones detalladas en la interfaz debe excluir la transacción T3 (del 05 de noviembre). 3. Usabilidad: El calendario y la selección de fechas deben ser intuitivos y funcionales (RNF-02). 4. Integración: Al cambiar el filtro, los datos cargados deben ser correctos, verificando la integración con el componente de consulta de transacciones.

Tabla 17.1 Diseño de caso de prueba de validación PV-003: Verificación del método configurarPerfilPersonal

Código	PV-003
Elemento a evaluar	Actualización de datos personales (nombre y contraseña) y la validación de seguridad que exige la contraseña actual, y volver a escribir la nueva contraseña para proceder con el cambio de contraseña.
Técnica de prueba	Caja Negra.
Descripción de la prueba	Se valida el flujo de acceso a la configuración del perfil y la capacidad de actualizar el nombre y la contraseña personal. Se enfoca en tres escenarios: 1) Éxito: Cambio de nombre y contraseña con la contraseña actual correcta. 2) Fallo: Intento cambiar la contraseña con una contraseña actual incorrecta. 3) Fallo: Intento cambiar la contraseña con la repetición de la nueva contraseña incorrecta.
Necesidades del entorno (Precondiciones)	Entorno de pruebas estándar. Un perfil personal existente con una contraseña conocida que pueda ser utilizada para la validación (Escenario de Éxito/Fallo). La interfaz de configuración de perfil debe ser accesible.

Tabla 17.2 Diseño de caso de prueba de validación PV-003: Verificación del método configurarPerfilPersonal

Especificación de entradas (Datos de prueba)	1. Usuario/Perfil: Sesión activa del perfil a modificar. 2. Datos Originales: Nombre ("Juan Pérez"), Contraseña Actual ("Clave123"). 3. Escenario de Éxito: a. Nuevo Nombre: "Juan P." b. Contraseña Actual: "Clave123" c. Nueva Contraseña: "NuevaClave789" d. Repetición Nueva Contraseña: "NuevaClave789" 4. Escenario de Fallo Contraseña Actual Incorrecta: a. Contraseña Actual (Incorrecta): "ClaveErronea" b. Nueva Contraseña: "NuevaClave789" 5. Escenario de Fallo Nuevas Contraseñas Diferentes: a. Nueva Contraseña: “NuevaClave123” b. Repetición Nueva Contraseña: “NuevaClaveDifent”
Especificación de salidas (Resultado esperado)	Escenario de Éxito: 1. El sistema debe mostrar un mensaje de confirmación ("Perfil actualizado con éxito"). 2. El nuevo nombre ("Juan P.") debe reflejarse en la interfaz del sistema. 3. Al intentar iniciar sesión nuevamente, solo la Nueva Contraseña ("NuevaClave789") debe permitir el acceso. Escenario de Fallo Contraseña Actual Incorrecta: 4. El sistema debe mostrar un mensaje de error específico ("La contraseña actual no es correcta"). 5. La contraseña debe mantenerse como "Clave123". Escenario de Fallo Nuevas Contraseñas Diferentes: 6. El sistema debe mostrar un mensaje de error específico ("Las contraseñas no coinciden").

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 18 Diseño de caso de prueba de sistema PS-001: Verificación de seguridad

Código	PS-001
Elemento a evaluar	Mecanismo de control de acceso (autorización) basado en roles (Administrador o Usuario Común).
Técnica de prueba	Caja Negra.
Descripción de la prueba	Se verifica que las funcionalidades críticas de administración (como ver el Balance Familiar - RF-08, o Eliminar un Perfil Familiar - RF-16) sean estrictamente inaccesibles para un usuario con el rol de Usuario Común.
Necesidades del entorno (Precondiciones)	Entorno de pruebas completo y estable. Se requiere que los roles de usuario y el mecanismo de validación de autorización en el backend están completamente implementados.
Especificación de entradas (Datos de prueba)	1. Datos de Base: Un usuario con rol Administrador (ej. "Admin1") y un usuario con rol Usuario Común (ej. "Comun1") deben estar registrados y validados (RF-01). 2. Acción de Prueba 1 (Fallo Esperado): Estar en la interfaz de Selección Perfil y tratar de navegar directamente a la URL de Balance (RF-08). 3. Acción de Prueba 2 (Fallo Esperado): Iniciar sesión como "Comun1" y tratar de navegar directamente a la URL de Eliminar Perfil Familiar (RF-16).
Especificación de salidas (Resultado esperado)	Escenario (Fallo Esperado): 1. RF-08 (Balance Familiar): Si se intenta acceder directamente a la URL, el sistema debe redirigir a una página de "Acceso Denegado". 2. RF-16 (Eliminar Perfil): Si el usuario común intenta acceder directamente a la URL, el sistema debe redirigir a una página de "Acceso Denegado". Criterio de Éxito: Las funciones críticas de administración son inaccesibles o bloqueadas para el rol Usuario Común.

Tabla 19 Diseño de caso de prueba de sistema PS-002: Verificación de usabilidad

Código	PS-002
Elemento a evaluar	Facilidad de uso, consistencia del diseño y eficiencia del sistema al ejecutar tareas básicas.
Técnica de prueba	Caja Negra (con observación del usuario).
Descripción de la prueba	Se verificará el tiempo y la tasa de éxito de un usuario nuevo o sin experiencia para completar una tarea fundamental (ej. registrar una transacción). Se evaluará la consistencia visual y la intuición del flujo de navegación (según el Mockup de la interfaz).
Necesidades del entorno (Precondiciones)	Entorno de pruebas estable. Se recomienda contar con una grabación de pantalla o un observador para capturar los puntos de fricción del usuario. El sistema debe tener un Usuario y Perfil de prueba previamente registrado para que el usuario pueda iniciar sesión.
Especificación de entradas (Datos de prueba)	1. Usuario de Prueba: Un individuo que nunca haya usado el sistema (idealmente un usuario final real). 2. Tarea Clave: El usuario debe completar el flujo de "Registrar un Ingreso Diario" (que involucra RF-04, RF-06) desde el inicio de sesión. 3. Métrica de Tiempo: El cronómetro se inicia al hacer clic en el botón de la tarea y se detiene al recibir el mensaje de confirmación de guardado.
Especificación de salidas (Resultado esperado)	1. Tiempo de Finalización: El tiempo promedio para completar la tarea de "Registrar un Ingreso Diario" debe ser menor a 120 segundos (< 2 min). 2. Tasa de Éxito: El usuario debe ser capaz de completar la tarea sin ayuda externa y sin errores de navegación el 90% de las veces (si se prueba con múltiples usuarios o repeticiones). 3. Consistencia: El diseño y la navegación deben ser coherentes con el estándar de la industria (ej. el botón de "Guardar" se encuentra donde se espera, la retroalimentación es inmediata).

8. Referencias

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