



Actividad 4 - Diseño del prototipo.

Código Banner: 100103874

Sneyder Cardona Mejía

Corporación Universitaria Iberoamericana

Proyecto de Software

Tatiana Lizbeth Cabrera Vargas

9 Noviembre 2025

Introducción

En un mundo donde la gestión financiera parece reservada para quienes tienen conexión estable y dispositivos avanzados, millones de personas quedan al margen. Este proyecto, desarrollado en el marco del semillero de investigación SIVFI, propone desarrollar e implementar una aplicación de escritorio autónoma que permita registrar y monitorear ingresos y gastos sin requerir internet.

A diferencia del enfoque de análisis y diseño, esta formulación avanza hacia la construcción de un MVP funcional, con persistencia local, interfaz navegable y despliegue real, todo ello pensado para quienes no pueden depender de la nube ni de sistemas complejos.

Dado que el proyecto forma parte de una línea de investigación institucional y por coherencia metodológica, se realiza de forma individual.

Contextualización de la necesidad

En muchos hogares colombianos, especialmente aquellos liderados por madres cabeza de familia o microempresarios informales, el control de las finanzas se hace de forma intuitiva: en libretas de cuadros, en notas del celular sin respaldo, o directamente en la memoria. Esta práctica, aunque funcional en el día a día, se vuelve frágil ante imprevistos, olvidos o cambios de rutina. Un gasto olvidado, una cuenta mal anotada o la pérdida de una libreta pueden desestabilizar todo el equilibrio financiero del mes.

El semillero de investigación SIVFI ha observado, a través de acompañamientos comunitarios y talleres de educación financiera, que una de las mayores barreras para una gestión presupuestal efectiva no es la falta de voluntad, sino la ausencia de herramientas accesibles que funcionen en sus condiciones reales: sin conexión estable a internet, sin dispositivos de alta gama, y sin tiempo para aprender sistemas complejos. Las aplicaciones disponibles en el mercado, aunque útiles en contextos urbanos conectados, resultan inalcanzables o irrelevantes para quienes viven en zonas con conectividad intermitente o nula.

Es en este espacio (entre la necesidad urgente de orden financiero y la imposibilidad de acceder a soluciones digitales tradicionales) donde surge la propuesta de un sistema de escritorio autónomo, diseñado para funcionar sin depender de la nube, con una interfaz clara, intuitiva y centrada en lo esencial: registrar, visualizar y tomar decisiones con lo que se tiene, aquí y ahora.

Planteamiento del problema

¿Cómo puede una madre cabeza de hogar llevar un control claro, confiable y autónomo de sus ingresos y gastos sin depender de una conexión a internet?

Objetivos generales y específicos.

Objetivo general

Desarrollar e implementar una aplicación de escritorio para el registro y seguimiento autónomo de ingresos y gastos sin conexión a internet, dirigida a madres cabeza de hogar y personas en contextos de conectividad limitada.

Objetivos específicos

Definir los requisitos funcionales y no funcionales del sistema, priorizando simplicidad, autonomía y usabilidad offline.

1. Construir un MVP funcional en Electron.js que permita registrar, consultar y monitorear transacciones financieras con persistencia local.
2. Implementar una interfaz navegable de alta fidelidad que guíe al usuario en cada acción sin requerir formación técnica.
3. Validar la funcionalidad básica mediante pruebas de usabilidad con el coordinador del semillero SIVFI.
4. Desplegar la aplicación en una plataforma gratuita (Render o GitHub Pages) y documentar el proceso de instalación.

Alcance del proyecto

El alcance de este proyecto incluye el análisis, diseño, desarrollo, pruebas y despliegue de una aplicación de escritorio funcional que permita el registro y seguimiento autónomo de ingresos y gastos sin conexión a internet. Se desarrollará un MVP (Producto Mínimo Viable) en Electron.js, con persistencia local mediante localStorage, interfaz navegable de alta fidelidad (diseñada en Figma) y despliegue en una plataforma gratuita como Render.

No se incluirá sincronización en la nube, análisis predictivo, autenticación de usuarios ni integración con servicios externos. El enfoque se mantiene en la simplicidad, autonomía y usabilidad offline, priorizando funcionalidades esenciales: registro de transacciones, resumen mensual y alerta de presupuesto.

Las herramientas utilizadas serán Figma (diseño de interfaces), Trello (gestión ágil con metodología Kanban), GitHub (control de versiones) y Render (despliegue). El tiempo estimado para completar esta fase es de 6 semanas, distribuidas en 3 sprints de 2 semanas cada uno.

Estructura del Desglose del Trabajo (EDT)

La EDT organiza el proyecto en fases, subfases y tareas específicas, permitiendo una planificación clara y realista. Dado que se trata de un MVP funcional desarrollado de forma individual, la estructura se enfoca en las etapas esenciales del ciclo de vida del desarrollo de software: análisis, diseño, desarrollo, pruebas y despliegue.

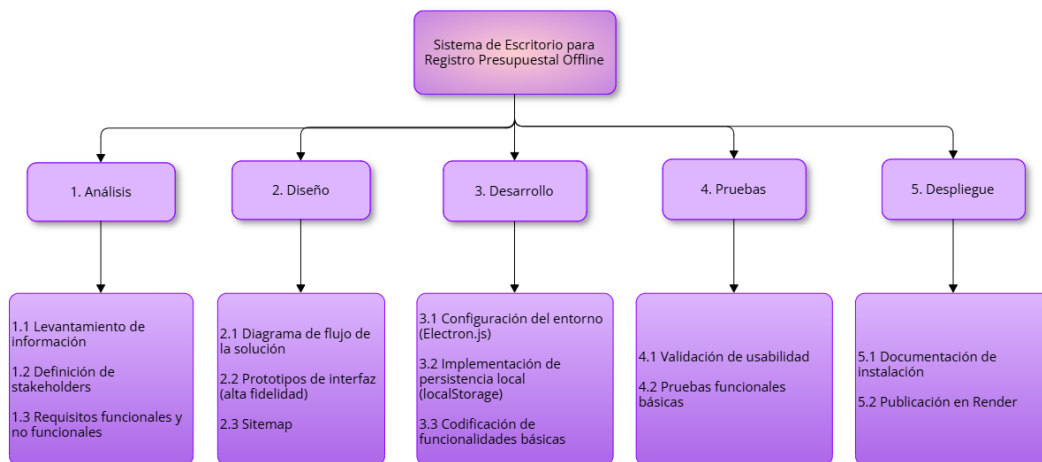


Figura 1. Estructura del Desglose del Trabajo (EDT)

Metodología ágil seleccionada

Para la gestión de las tareas de formulación, desarrollo y despliegue del proyecto, se adopta la metodología Kanban, por su flexibilidad, simplicidad y enfoque visual, ideal para un equipo de una sola persona. Esta metodología permite visualizar el flujo de trabajo en columnas

(To Do, In Progress, Done) y mover las tareas a medida que avanzan, sin la rigidez de sprints o roles definidos como en Scrum.

El tablero con todas las tareas del proyecto está disponible en:

<https://trello.com/invite/b/68e2d138ed5d0dad820d0c20/ATTI8f69112e1a7aea32061c47269ca58f64E1801DC5/proyecto-de-software-sistema-presupuestal-offline>

Adicionalmente, todos los artefactos del proyecto (diagramas, matriz de riesgos, documento final, etc.) se almacenan en el repositorio de GitHub:

<https://github.com/SneyderCM/ProyectoDeSoftware.git>

Justificación

Este proyecto no busca revolucionar la tecnología, sino devolver el control a quienes más lo necesitan. A corto plazo, entregará un MVP funcional una aplicación de escritorio autónoma que permita registrar ingresos y gastos, visualizar saldos y recibir alertas sin depender de internet, validado con el coordinador del semillero SIVFI.

A mediano plazo, este MVP puede convertirse en una herramienta real para madres cabeza de hogar y microempresarios informales, especialmente en zonas rurales o marginadas donde la conectividad es intermitente o inexistente. Su simplicidad, enfoque offline y diseño centrado en lo esencial lo hacen accesible incluso para usuarios sin formación técnica.

A largo plazo, si se replica o adapta, podría contribuir a la inclusión financiera en contextos de baja conectividad, empoderando a comunidades que viven al margen de la nube, pero no del derecho a decidir con información.

El verdadero “plus” de esta propuesta no está en algoritmos ni en inteligencia artificial, sino en respetar la realidad del usuario: sin exigir lo que no tiene, sin sobrecargarlo con funciones innecesarias, y sin asumir que todos viven conectados. Para las madres que anotan sus cuentas en libretas desgastadas, esto significa dignidad y autonomía. Para el semillero SIVFI, coherencia con su misión social. Y para la universidad, un ejemplo de tecnología con propósito, no con pretensiones.

Por eso, invertir tiempo y esfuerzo en este diseño no es un ejercicio académico: es sembrar una solución que, aunque modesta, puede ayudar a alguien a mirar su mes con un poco más de calma... y un poco menos de incertidumbre.

Mapa de Stakeholders y clasificación

Los stakeholders del proyecto se identificaron y clasificaron según su nivel de influencia e interés, utilizando un mapa de cuadrantes. El coordinador del semillero SIVFI, por su rol directo en la validación de la necesidad, se ubica en el cuadrante de alta influencia y alto interés, por lo que se gestionará de cerca. Las madres cabeza de hogar, como usuarios finales, tienen alto interés, pero baja influencia, por lo que se mantendrán informadas. La profesora y la universidad, aunque con alta influencia institucional, tienen un interés limitado al cumplimiento académico,

por lo que se mantendrán satisfechas. Finalmente, las herramientas tecnológicas (Figma, Trello, GitHub) se monitorean como recursos de apoyo.

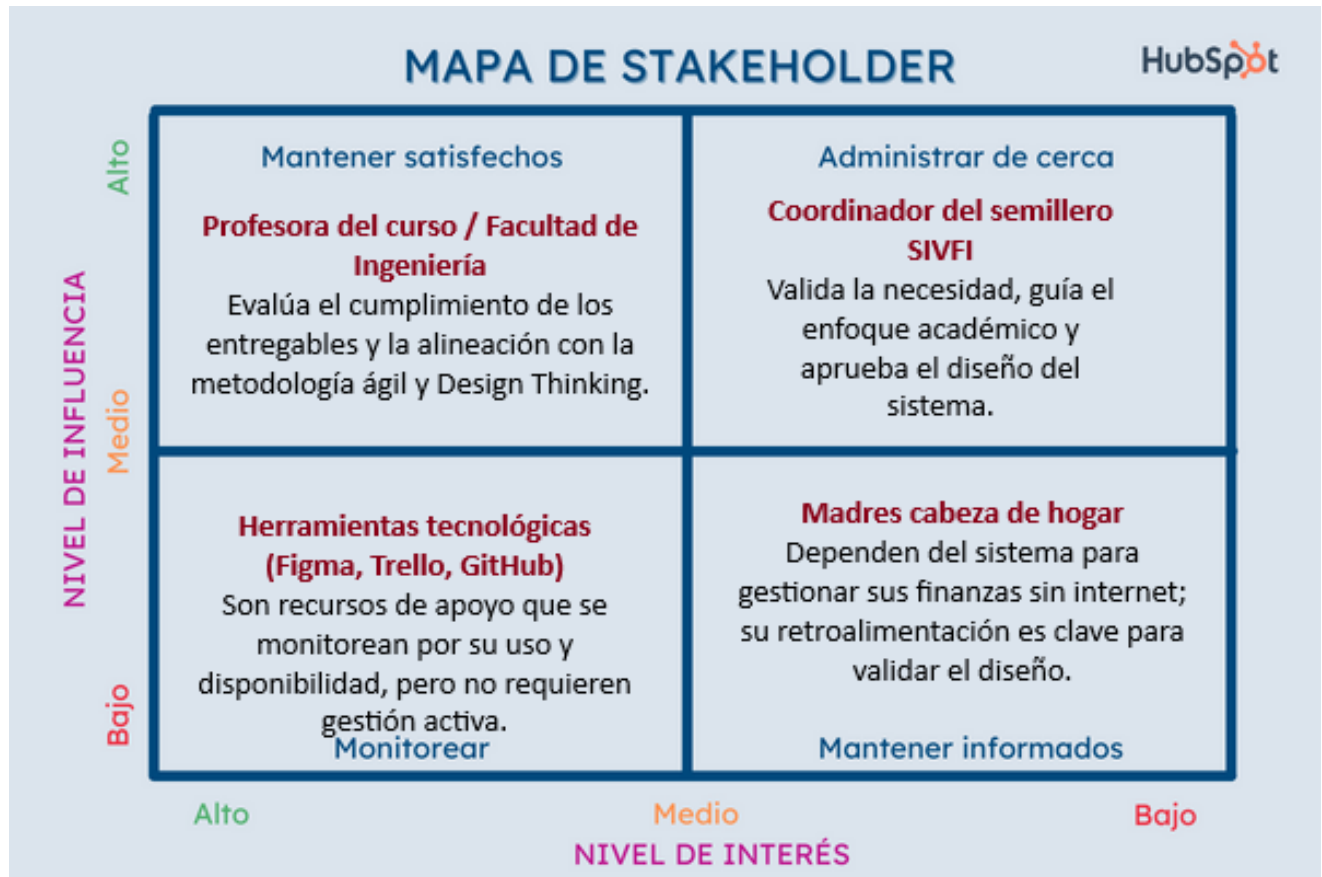


Figura 2. Mapa de Stakeholders

Matriz de riesgos

Durante la fase de formulación del proyecto, se identificaron los principales riesgos asociados al desarrollo del MVP. Estos riesgos se derivan directamente de los requisitos funcionales y del contexto de desarrollo individual. La matriz a continuación presenta cada riesgo, su probabilidad, impacto y la estrategia definida para mitigarlo.

Riesgo	Probabilidad	Impacto	Estrategia de Mitigación
Dificultad para implementar persistencia local con localStorage debido a la falta de experiencia en desarrollo frontend.	Media	Alto	Usar tutoriales de YouTube y documentación oficial de Electron.js. Validar con ejemplos simples antes de integrar funcionalidades completas.
Falla en el despliegue del MVP en Render por configuración incorrecta del entorno.	Media	Medio	Seguir guías paso a paso de Render para aplicaciones Electron.js. Hacer pruebas locales exhaustivas antes del despliegue.
Sobrecomplejación del código al intentar añadir funcionalidades avanzadas (ej. exportar a PDF, gráficos).	Alta	Alto	Enfocarse estrictamente en el módulo básico: registro de transacciones, resumen mensual y alerta de presupuesto. Revisar semanalmente el alcance definido.
Falta de retroalimentación real del usuario (madres cabeza de hogar) para validar la usabilidad.	Media	Alto	Validar el MVP con el coordinador del semillero SIVFI, quien tiene contacto directo con el público objetivo, y usar escenarios de uso realistas.

Figura 2. Matriz de riesgos

Presupuesto

El presupuesto estimado para la formulación, desarrollo y despliegue del MVP se basa en un esfuerzo individual de 60 horas, con un valor promedio de \$10.000 COP por hora (estándar para prácticas académicas en desarrollo de software en Colombia). Las herramientas utilizadas son gratuitas, y el despliegue se realiza en una plataforma de código abierto sin costo.

Semillero SIVFI

Universidad de Bogotá, Carrera 7 # 40-50, Bogotá D.C.

Factura

Fecha: 2025-10-05

Nº de factura: 1

Facturar a

Semillero SIVFI

Universidad de Bogotá, Carrera 7 # 40-50,

Bogotá D.C.

Cant.	Descripción	Precio unitario	Total
60	Mano de obra (60 horas de desarrollo)	\$10,000	\$600,000
1	Uso de herramientas gratuitas (Figma, GitHub, Trello, Render)	\$0	\$0
1	Despliegue en plataforma gratuita (Render)	\$0	\$0

Total

\$600,000

Saldo

\$600,000

Este presupuesto es estimado y corresponde a un proyecto académico de desarrollo de software. La mano de obra se calcula en 60 horas a \$10.000 COP/hora, valor promedio para estudiantes en prácticas. Las herramientas y el despliegue son gratuitos.

Gracias por su preferencia.

Figura 3. Presupuesto

Levantamiento de información

Para comprender a fondo la necesidad de gestión financiera en contextos de baja conectividad, se realizó un levantamiento de información mediante un método interactivo, específicamente una entrevista semiestructurada al coordinador del semillero SIVFI, quien tiene contacto directo con madres cabeza de hogar.

Método: Interactivo

Técnica: Entrevista semiestructurada

Instrumento: Guion con 5 preguntas clave:

- ¿Cómo gestionan actualmente sus finanzas las personas con las que trabaja el semillero?
- ¿Qué herramientas utilizan (libretas, apps, memoria)?
- ¿Cuáles son sus principales dificultades para llevar un control presupuestal?
- ¿Tienen acceso constante a internet?
- ¿Qué características debería tener una herramienta ideal para ellos?

Conclusiones:

Los hallazgos confirman que la mayoría depende de libretas físicas o memoria, lo que genera pérdida de información y estrés financiero. Las apps existentes no son viables por la falta de conexión estable. La necesidad prioritaria es una herramienta offline, visual, simple y centrada en lo esencial: registrar gastos, ver saldos y recibir alertas sin teclear mucho. Esta validación respalda el enfoque del proyecto: un sistema de escritorio autónomo, sin dependencia de la nube.

Diagrama de flujo

El diagrama de flujo representa el recorrido principal que seguirá una madre cabeza de hogar al interactuar con el sistema. Inicia al abrir la aplicación, donde visualiza su saldo actual y presupuesto mensual. Desde allí, puede optar por registrar una nueva transacción (ingreso o gasto), ingresar los datos correspondientes y recibir retroalimentación inmediata: confirmación de guardado o alerta si se acerca al límite del presupuesto. Este flujo prioriza la simplicidad, la autonomía y la claridad visual, evitando pasos innecesarios o tecleo excesivo.

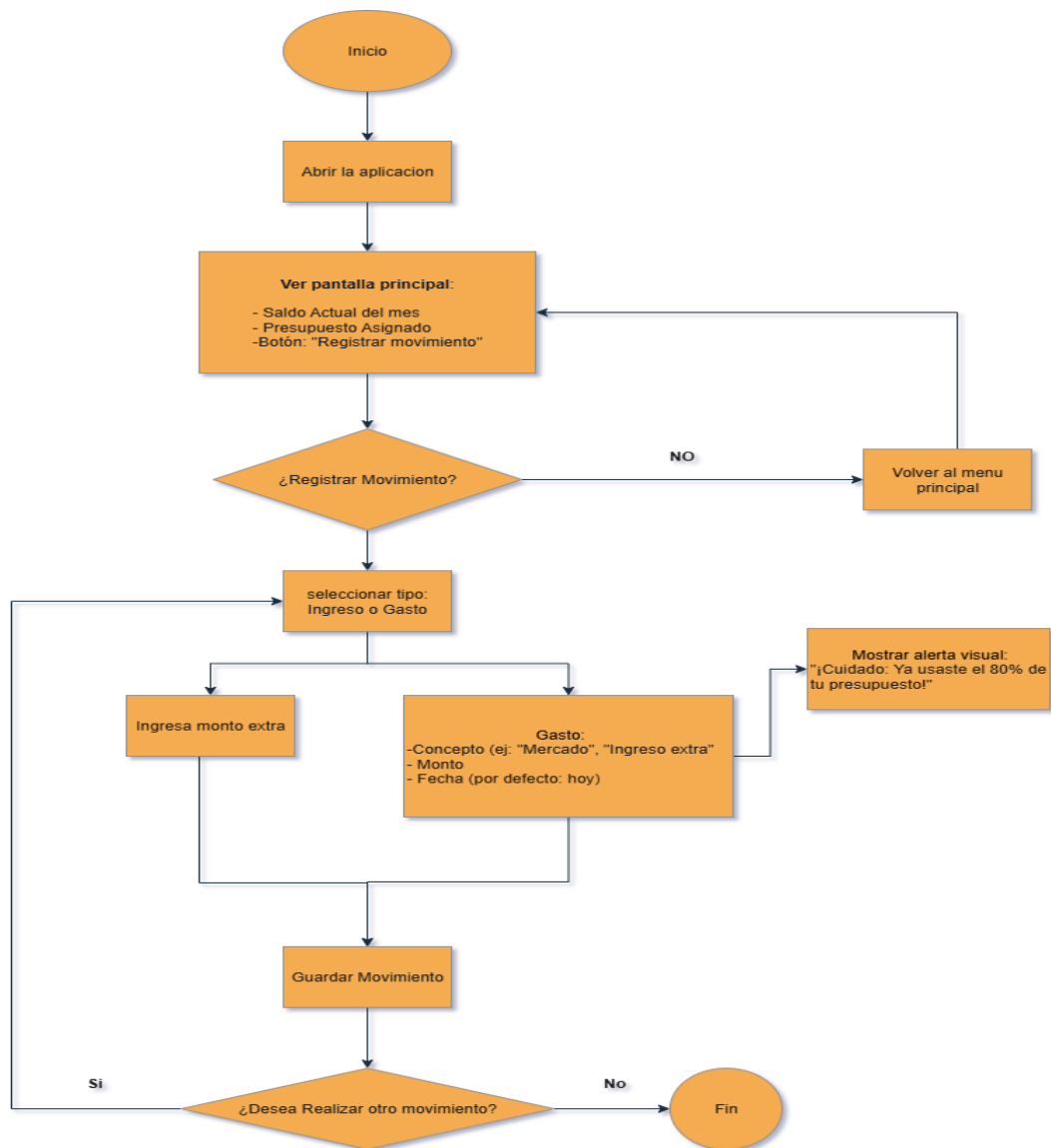


Figura 3. Diagrama de flujo

Historias de usuario:

Historia de usuario 1: Como madre cabeza de hogar, quiero registrar mis gastos diarios, para llevar un control claro de mis egresos sin depender de internet.

Criterios de aceptación:

Escenario feliz:

- Ingreso concepto (ej. “Mercado”), monto (\$50.000) y fecha (hoy).
- El sistema guarda la transacción localmente y muestra mensaje: *“Gasto registrado”*.

Escenario no feliz:

- Si no ingreso monto, el sistema muestra: *“El monto es obligatorio”*.
- Si ingreso un monto no numérico, el sistema muestra: *“Por favor, ingrese un número válido”*.

Historia de usuario 2: Como microempresario informal, quiero consultar un resumen de mis ingresos y gastos del mes actual, para saber si estoy generando ganancias o pérdidas.

Criterios de aceptación:

Escenario feliz:

Al abrir la pantalla de resumen, veo:

- Total ingresos: \$1.200.000
- Total gastos: \$900.000
- Saldo: \$300.000

Escenario no feliz:

- Si no hay transacciones registradas, el sistema muestra: “No hay movimientos este mes”.

Definición de Requisitos Funcionales y No Funcionales (RQF / RQNF)

Requisitos Funcionales (RQF)

CÓDIGO	REQUISITOS
RQF001	Nombre: Registrar transaccion
	Descripción: Permitir al usuario ingresar concepto, monto, tipo (ingreso/gasto) y fecha.
	Usuarios: Madres cabeza de hogar, microempresarios

CÓDIGO	REQUISITOS
RQF002	Nombre: Consultar resumen mensual
	Descripción: Mostrar total de ingresos, gastos y saldo del mes seleccionado.
	Usuarios: Madres cabeza de hogar, microempresarios

CÓDIGO	REQUISITOS
RQF003	Nombre: Establecer presupuesto mensual
	Descripción: Permitir definir un límite de gasto por mes.
	Usuarios: Madres cabeza de hogar, microempresarios

CÓDIGO	REQUISITOS
RQF004	Nombre: Recibir alerta de presupuesto
	Descripción: Notificar visualmente cuando el gasto alcance el 80% del presupuesto asignado.
	Usuarios: Madres cabeza de hogar, microempresarios

CÓDIGO	REQUISITOS
RQF005	Nombre: Editar o eliminar transacción
	Descripción: Permitir modificar o borrar una transacción existente.
	Usuarios: Madres cabeza de hogar, microempresarios

Requisitos No Funcionales (RQNF):

CÓDIGO	REQUISITOS
RQNF001	Nombre: Usabilidad
	Descripción: La interfaz debe ser intuitiva para usuarios sin formación técnica (máximo 2 clics para registrar un gasto).

CÓDIGO	REQUISITOS
RQNF002	Nombre: Autonomía offline
	Descripción: El sistema debe funcionar sin conexión a internet; todos los datos se almacenan localmente.

CÓDIGO	REQUISITOS
RQNF003	Nombre: Tiempo de respuesta
	Descripción: Las acciones deben responder en menos de 1 segundo.

CÓDIGO	REQUISITOS
RQNF004	Nombre: Accesibilidad visual
	Descripción: Uso de colores con alto contraste y texto legible (tamaño mínimo 14pt).

Evidencias de herramientas de Design Thinking

1. Empatizar

Herramienta utilizada: Entrevista semiestructurada + Mapa de empatía

Con quién: Coordinador del semillero SIVFI (cliente real que tiene contacto directo con madres cabeza de hogar)

Cómo se aplicó: Se realizó una entrevista con 5 preguntas clave sobre hábitos financieros, herramientas usadas y principales dificultades. Con esa información, se construyó un mapa de empatía para comprender lo que el usuario dice, piensa, siente y hace.

Entregables derivados: Contextualización de la necesidad, planteamiento del problema, justificación.

2. Definir

Herramienta utilizada: Lluvia de ideas + Análisis de causa-efecto

Con quién: Trabajo individual (por ser proyecto de semillero)

Cómo se aplicó: A partir de los hallazgos de la fase de empatía, se generó una lista de posibles problemas y se analizaron sus causas raíz (ej. “pérdida de control financiero” → causas: sin internet, sin formación, sin herramientas simples). Esto permitió delimitar el problema central y establecer los límites del proyecto.

Entregables derivados: Objetivos, alcance, metodología ágil, mapa de stakeholders.

3. Idear

Herramienta utilizada: Matriz de prioridades + Brainstorming

Con quién: Trabajo individual

Cómo se aplicó: Se generaron múltiples ideas de funcionalidades (ej. alerta de presupuesto, resumen mensual, exportar a PDF) y se evaluaron mediante una matriz que

consideró impacto en el usuario y viabilidad técnica. Solo se priorizaron las funcionalidades más esenciales y realistas para un desarrollo individual.

Entregables derivados: Requisitos funcionales, historias de usuario, diagrama de flujo de la solución.

Prototipo funcional

Se ha desarrollado un prototipo funcional mínimo (MVP) en HTML, CSS y JavaScript puro, que permite a la usuaria:

- Asignar un presupuesto mensual fijo (usado únicamente como límite para gastos).
- Registrar ingresos (tratados como ganancias, sin afectar el presupuesto).
- Registrar gastos, los cuales se descuentan del presupuesto asignado.
- Visualizar un resumen informativo con:
 - Presupuesto asignado
 - Gastos acumulados
 - Saldo actual = Presupuesto – Gastos
 - Ingresos totales (solo para contexto)
- Recibir una alerta visual en color naranja cuando los gastos alcancen o superen el 80% del presupuesto asignado.

El sistema funciona 100% offline, utilizando localStorage para persistir los datos en el navegador del usuario, garantizando autonomía en contextos sin conexión a internet.

Enlace del prototipo funcional (desplegado):

<https://sneydercm.github.io/presupuestal.mvp/>

Instrucciones de uso:

Abrir el enlace en cualquier navegador moderno (Chrome, Edge, Firefox). No requiere instalación ni conexión a internet. Para pruebas repetidas, se incluye un botón “Reiniciar todo” que restablece el estado del sistema.

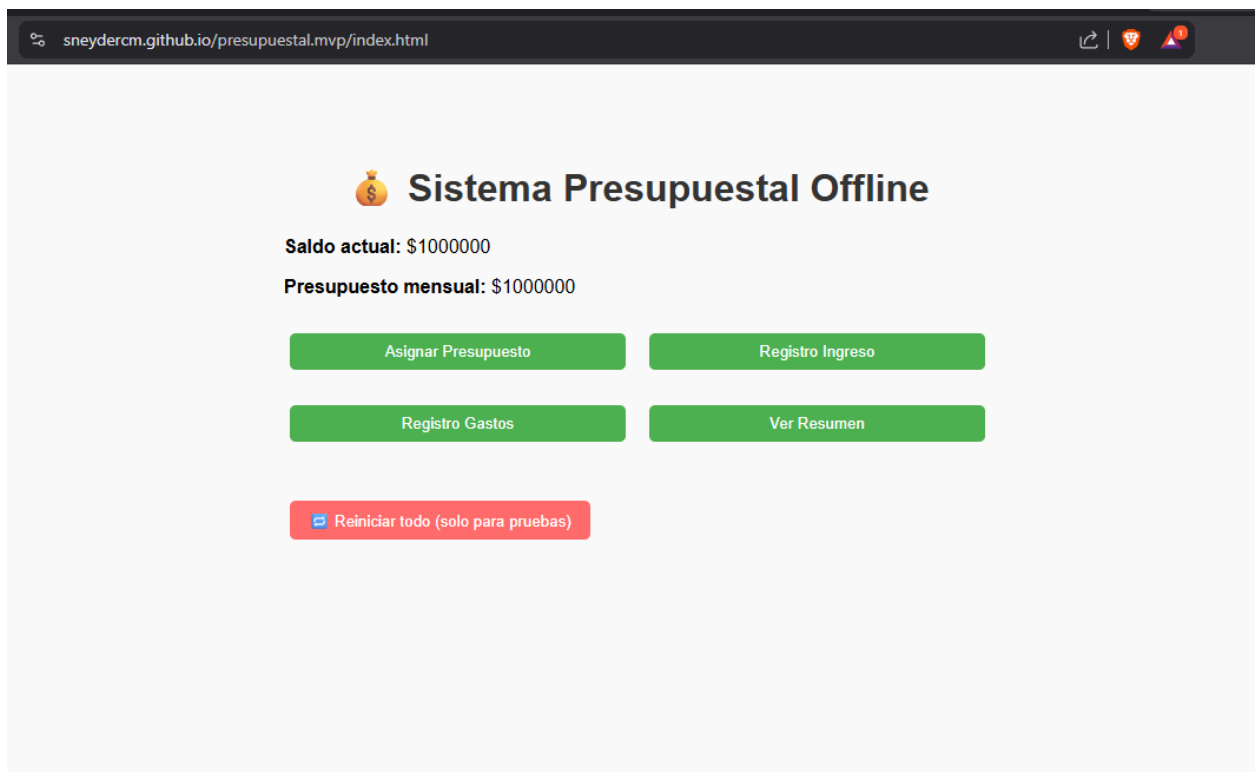


Figura 4. Pantalla principal del sistema

sneydercm.github.io/presupuestal.mvp/registrar.html?tipo=gasto

+ Registrar Gasto o Ingreso

Tipo:

Monto:

Fecha (por defecto: hoy):

Descripción:

Figura 5. Formulario de registro de transacciones



Figura 6. Pantalla de resumen con alerta del 80% activada

Repositorio en GitHub: <https://github.com/SneyderCM/presupuestal.mvp.git>

Modelos de Comportamiento y Estructura

Casos de uso:

CU-01: Asignar Presupuesto Mensual

- Actor: Madre cabeza de hogar, estudiante
- Descripción: El usuario define un monto fijo para su presupuesto mensual, usado exclusivamente como límite de control para gastos.
- Flujo principal:
 1. El usuario accede a la aplicación.
 2. Selecciona “Asignar Presupuesto”.

3. Ingresa un monto válido ($> \$0$).
 4. Confirma la acción.
 5. El sistema guarda el valor y lo muestra en la pantalla principal.
- Flujo alternativo:
 - Si el monto es inválido, el sistema muestra: *“Por favor ingrese un monto mayor a \$0”*.
 - Precondiciones: Ninguna.
 - Poscondiciones: El presupuesto queda establecido como referencia para gastos.
 - Frecuencia: Baja.

CU-02: Registrar Gasto

- Actor: Madre cabeza de hogar, estudiante
- Descripción: El usuario registra un gasto, que se descuenta del presupuesto asignado y puede activar una alerta visual al alcanzar el 80%.
- Flujo principal:
 1. El usuario selecciona “Registro Gastos”.
 2. Ingresa monto, descripción y fecha.
 3. Confirma el registro.
 4. El sistema guarda la transacción.
 5. Actualiza el saldo ($\text{Saldo} = \text{Presupuesto} - \text{Gastos}$).
 6. Si los gastos $\geq 80\%$ del presupuesto, activa alerta naranja en el resumen.
- Flujo alternativo:
 - Si el monto es inválido, el sistema muestra: *“El monto debe ser mayor a \$0”*.

- Precondiciones: Recomendado haber asignado presupuesto.
- Poscondiciones: Transacción registrada; saldo y alerta actualizados.
- Frecuencia: Alta.

CU-03: Registrar Ingreso

- Actor: Madre cabeza de hogar, estudiante
- Descripción: El usuario registra un ingreso (tratado como “ganancia”), que no afecta el presupuesto, solo se muestra en el resumen informativo.
- Flujo principal:
 1. El usuario selecciona “Registro Ingreso”.
 2. Ingresa monto, descripción y fecha.
 3. Confirma el registro.
 4. El sistema guarda la transacción y actualiza el total de ingresos en el resumen.
- Flujo alternativo:
 - Si el monto es inválido, el sistema muestra: *“El monto debe ser mayor a \$0”*.
- Precondiciones: Ninguna.
- Poscondiciones: Ingreso registrado y visible en el resumen.
- Frecuencia: Media.

CU-04: Consultar Resumen Mensual

- Actor: Madre cabeza de hogar, estudiante
- Descripción: El usuario visualiza un resumen informativo con presupuesto, gastos, saldo (presupuesto – gastos), ingresos y estado de la alerta del 80%.

- Flujo principal:
 1. El usuario selecciona “Ver Resumen”.
 2. El sistema muestra:
 - Presupuesto asignado
 - Gastos acumulados
 - Saldo actual
 - Ingresos totales
 - Alerta naranja (si aplica)
 - Últimas transacciones
- Flujo alternativo:
 - Si no hay transacciones, muestra: *“No hay movimientos registrados”*.
- Precondiciones: Ninguna.
- Poscondiciones: El usuario obtiene visibilidad clara de su estado financiero.
- Frecuencia: Alta.

UML

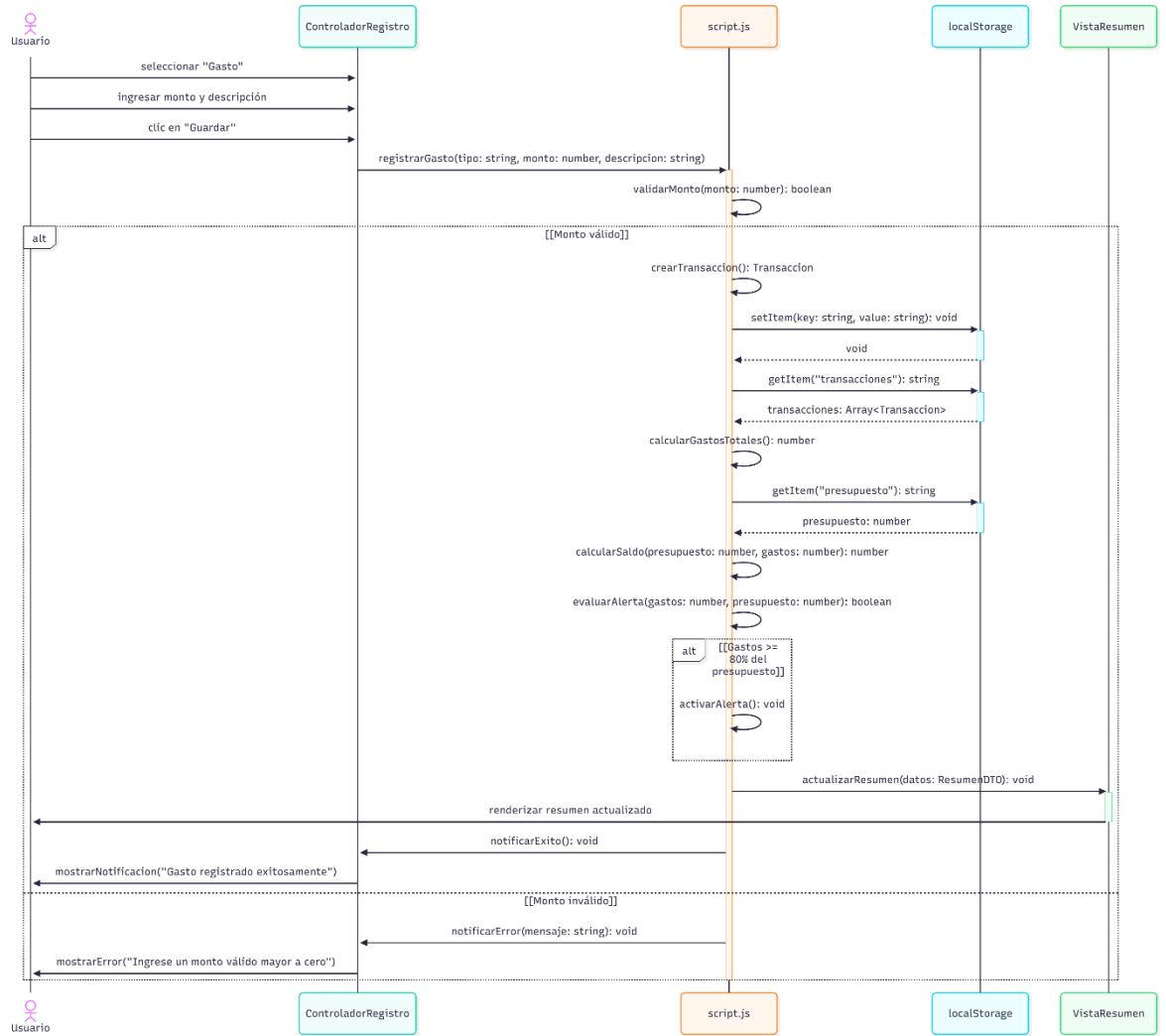


Figura 7. Diagrama secuencial

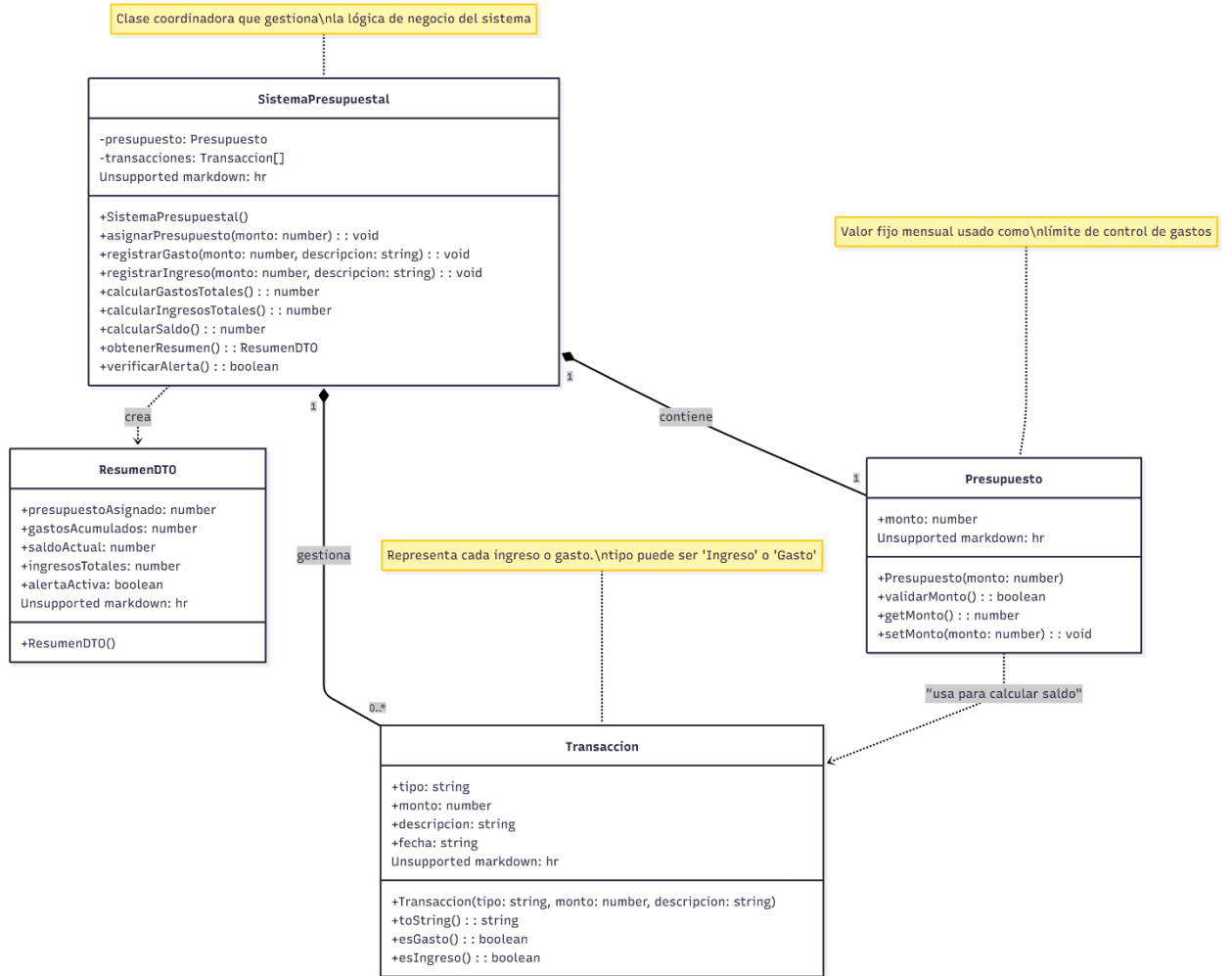


Figura 8. Diagrama de clases.

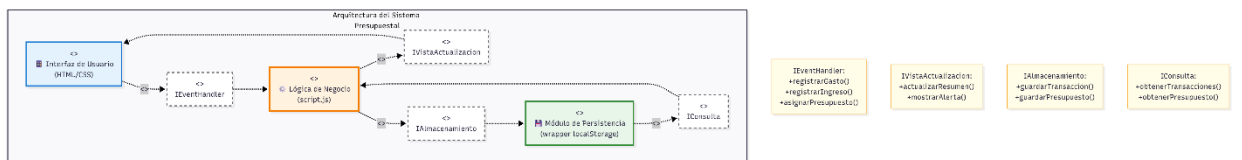


Figura 9. Diagrama componentes

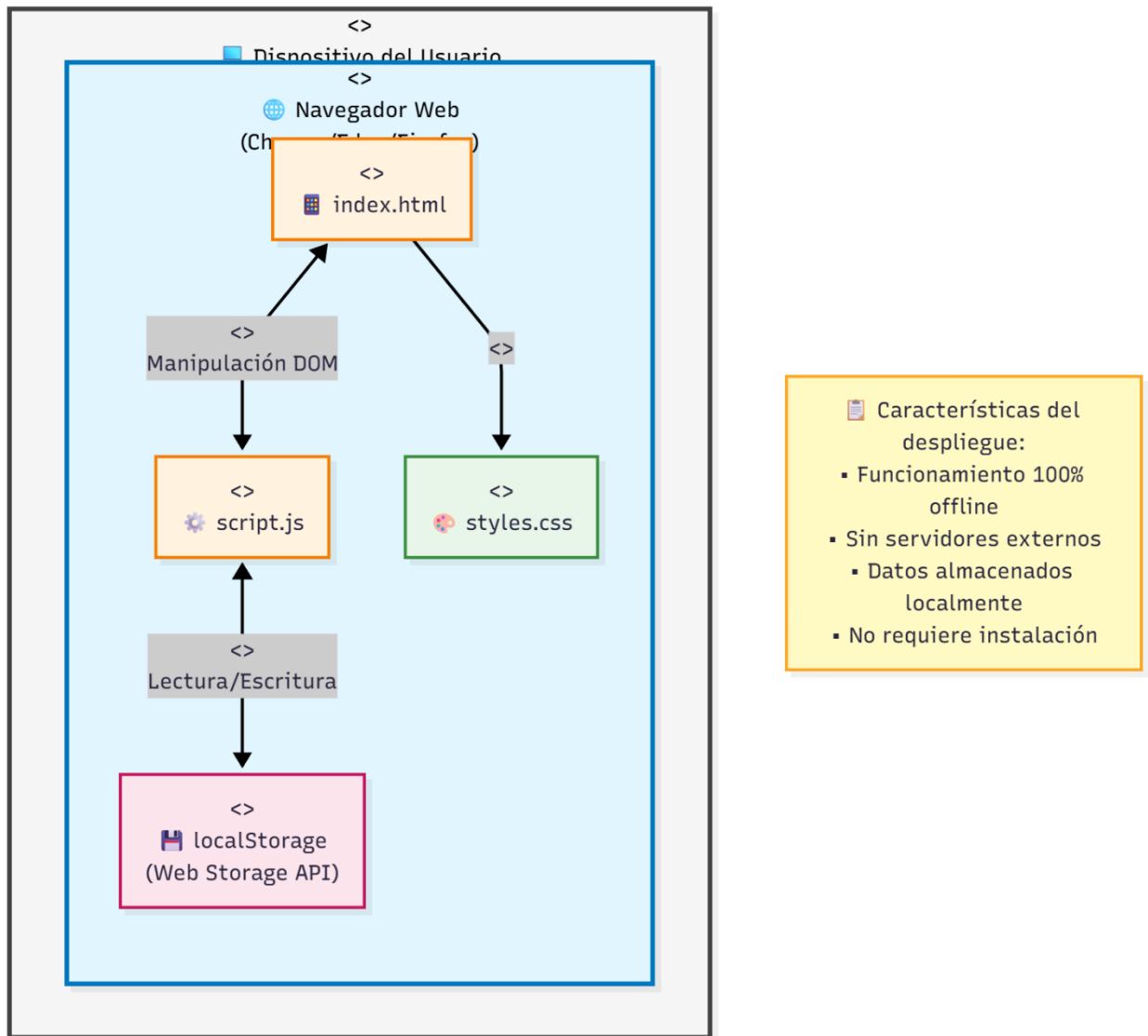


Figura 10. Diagrama despliegue.

Modelo de base de datos

El sistema guarda toda la información directamente en el navegador de la usuaria, usando una herramienta que se llama localStorage. Esto significa que no necesita internet ni servidores externos para funcionar. Es ideal para madres cabeza de hogar o estudiantes que no siempre tienen conexión estable, porque todo lo que registran queda guardado en su propio dispositivo.

La información se organiza en dos partes principales:

- Presupuesto: un número único que representa el monto total que la usuaria decide gastar en el mes.
- Transacciones: una lista de todos los movimientos que registra, ya sean ingresos o gastos.

Cada movimiento incluye:

- si es ingreso o gasto,
- el monto,
- una breve descripción (por ejemplo, “mercado” o “pago de colegio”),
- y la fecha.

Gracias a este enfoque, los usuarios pueden usar la aplicación en cualquier momento, desde cualquier lugar, sin depender de internet ni de herramientas complejas. Todo está pensado para ser sencillo, seguro y 100% offline

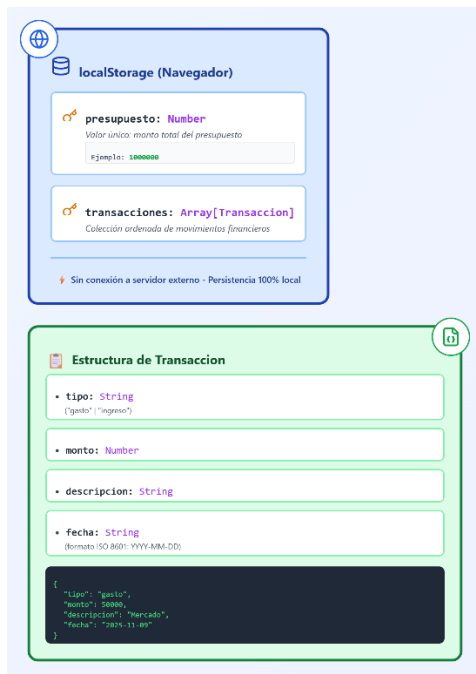


Figura 11. Diagrama de arquitectura de base de datos NoSQL basada en `localStorage`

Prototipos de interfaz

Prototipo de baja fidelidad.

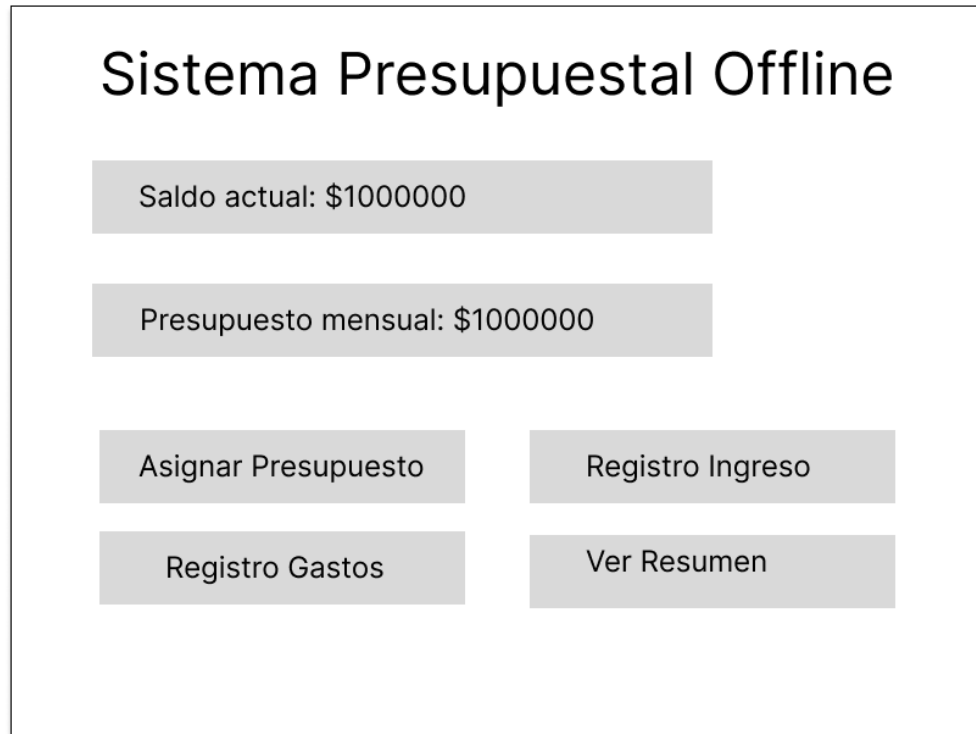


Figura 11. Baja-inicio

Asignar presupuesto mensual

Asignar Presupuesto

\$1000000

Guardar

Figura 12. Baja-asignar presupuesto

Registrar Gasto o Ingreso

Tipo: Ingreso

Monto: \$50000

Fecha: 22 - 04 -2025

Descripcion: Gasolina semanal

Guardar Cancelar

Figura 13. Baja-formulario

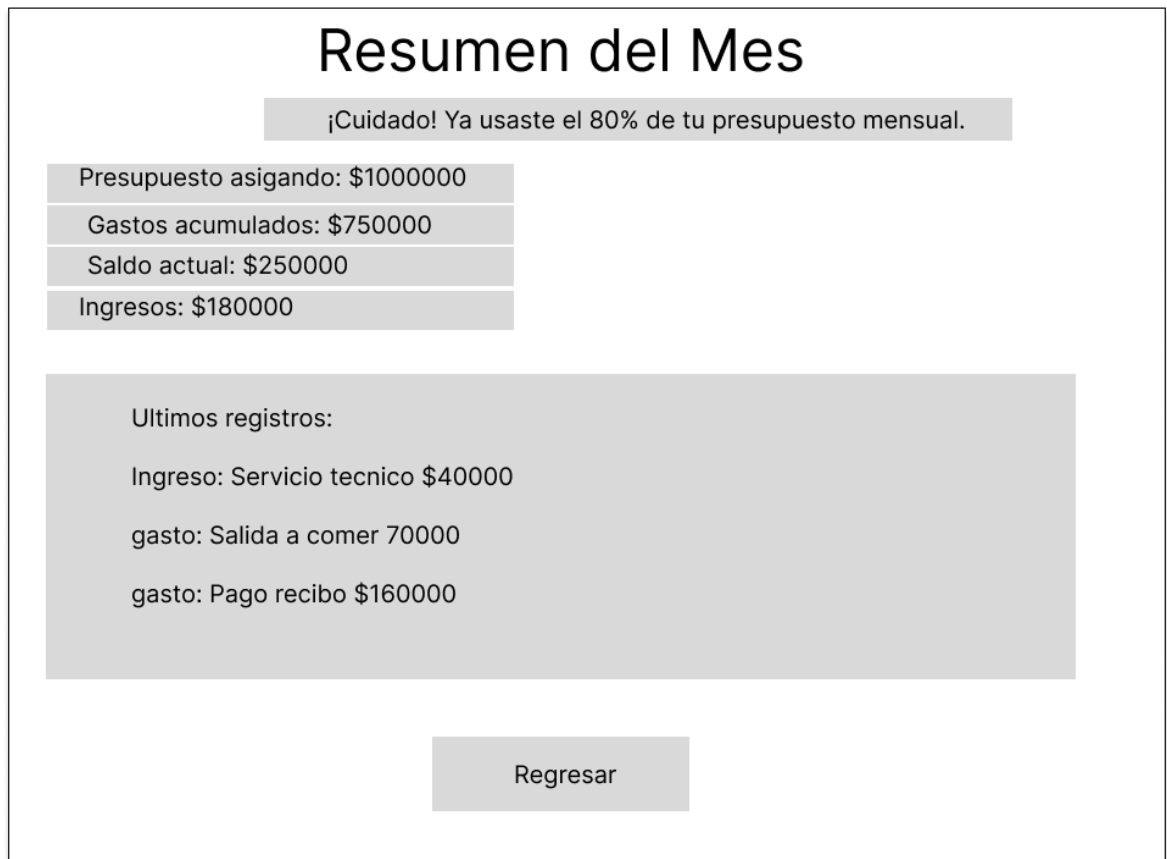


Figura 14. Baja – resumen

Enlace Figma: <https://www.figma.com/design/fFLJI3qkvEGeIAHBRBHtSX/Prototipos-de-baja-fidelidad---Proyecto?node-id=26-132&t=9okuv8oOgjB1a27C-0>

Prototipo de Alta fidelidad.



Sistema Presupuestal Offline

Saldo actual: \$1000000

Presupuesto mensual: \$1000000

Asignar Presupuesto

Registro Ingreso

Registro Gastos

Ver Resumen

Figura 15. Alta-inicio



Asignar presupuesto mensual

Asignar Presupuesto

\$1000000

Guardar

Figura 16. Alta-asignar presupuesto

+ Registrar Gasto o Ingreso

Tipo: Ingreso

Monto: \$50000

Fecha: 22 - 04 -2025

Descripcion: Gasolina semanal

Guardar

Cancelar

Figura 17. Alta-formulario



Figura 18. Alta-resumen

Pruebas de usabilidad

Se realizaron pruebas de usabilidad no moderadas con el prototipo funcional desplegado en GitHub Pages, con el objetivo de validar la claridad del flujo de usuario, la intuitividad de las interfaces y la comprensión de la alerta presupuestal. La prueba se aplicó a 5 participantes representativos del público objetivo: 3 madres cabeza de hogar y 2 estudiantes universitarios, cada uno de los cuales realizó las 5 tareas clave del sistema.

Detalles de la prueba:

- Nombre del proyecto: Sistema Presupuestal Offline

- Versión del prototipo: v1.0
- Fecha de la prueba: 29 de Octubre de 2025
- Equipo responsable: Sneyder Cardona
- Herramientas usadas: Prototipo funcional en GitHub Pages, formulario de observación manual
- Objetivos de la prueba:
 - Validar que el flujo de registro de transacciones sea intuitivo.
 - Confirmar que la alerta del 80% sea clara y visible.
 - Verificar que el resumen mensual sea informativo y comprensible.
- Número de participantes: 5

Pruebas de usabilidad – María Issabel Vasquez

Rol: Madre cabeza de hogar (36 años)

Fecha: 29 de Octubre de 2025

Herramienta usada: Prototipo funcional en GitHub Pages

Tarea / Escenario	¿Completada?	Observaciones
1. Asignar un presupuesto mensual de \$1.000.000	Sí	Encontró el botón "Asignar Presupuesto" fácilmente en la pantalla principal. Al hacer clic, ingresó el monto sin dudar. Comentó: "Esto es sencillo, me gusta que sea directo."
2. Registrar un gasto de \$300.000 con la descripción "Mercado"	Sí	Seleccionó correctamente la opción "Gasto", ingresó el monto y la descripción sin ayuda. Dijo: "Listo, ya quedó guardado."
3. Registrar un ingreso de \$200.000 con la descripción "Venta de ropa"	Sí	Reconoció la diferencia entre ingreso y gasto. Al ver el saldo actual, preguntó: "¿Entonces lo que gano no se suma al presupuesto?" y al recibir la explicación, respondió: "Perfecto, así no me confundo."
4. Registrar un gasto adicional de \$500.000 para alcanzar el 80% del presupuesto	Sí	Ingresó el gasto sin dificultad. Al regresar al resumen, notó inmediatamente la alerta en naranja y dijo: "¡Ah, ya me avisó que voy por el 80%! Eso me da seguridad."
5. Consultar el resumen mensual y explicar qué muestra	Sí	Identificó correctamente todos los elementos: presupuesto asignado (\$1.000.000), gastos totales (\$800.000), saldo actual (\$200.000) e ingresos (\$200.000). Añadió: "La alerta en naranja me dice cuándo debo parar de gastar."

Figura 18. Tabla Maria

Pruebas de usabilidad – Duvan Felipe

Rol: Estudiante universitario (22 años)

Fecha: 29 de Octubre de 2025

Herramienta usada: Prototipo funcional en GitHub Pages

Tarea / Escenario	¿Completada?	Observaciones
1. Asignar un presupuesto mensual de \$1.000.000	Sí	Navegó directamente a “Asignar Presupuesto” y lo completó sin ayuda. Comentó: “Está claro, no hay que adivinar.”
2. Registrar un gasto de \$300.000 con la descripción “Mercado”	Sí	Seleccionó “Gasto” sin dudar, llenó todos los campos. Dijo: * “El formulario es intuitivo, no tiene elementos de más.”
3. Registrar un ingreso de \$200.000 con la descripción “Venta de ropa”	Sí	Cambió el tipo a “Ingreso” y completó la descripción. Preguntó: “¿Esto no afecta el presupuesto, cierto?” y al confirmársele, respondió: “Bien, así lo entiendo.”
4. Registrar un gasto adicional de \$500.000 para alcanzar el 80% del presupuesto	Sí	Ingresó el monto y, al ir al resumen, identificó la alerta naranja de inmediato. Dijo: “¡Ahí está la alerta! Me dice que ya gasté mucho.”
5. Consultar el resumen mensual y explicar qué muestra	Sí	Explicó correctamente: “Presupuesto: \$1.000.000, gastos: \$800.000, saldo: \$200.000, ingresos: \$200.000. Y la alerta me dice que voy por el 80%.”

Figura 19. Tabla Duvan

Pruebas de usabilidad – Ana Martínez

Rol: Madre cabeza de hogar (38 años)

Fecha: 29 de Octubre de 2025

Herramienta usada: Prototipo funcional en GitHub Pages

Tarea / Escenario	¿Completada?	Observaciones
1. Asignar un presupuesto mensual de \$1.000.000	Sí	Tardó unos segundos en encontrar el botón, pero lo logró. Dijo: “Está bien ubicado, no se esconde.”
2. Registrar un gasto de \$300.000 con la descripción “Mercado”	Sí	Completó el formulario sin errores. Añadió: “Poner la descripción me ayuda a recordar en qué gasté.”
3. Registrar un ingreso de \$200.000 con la descripción “Venta de ropa”	Sí	Reconoció la diferencia con el gasto. Comentó: “Me gusta que los ingresos no se mezclen con el presupuesto; así sé cuánto tengo para gastar.”
4. Registrar un gasto adicional de \$500.000 para alcanzar el 80% del presupuesto	Sí	Al ver la alerta, dijo: “Esto es muy útil. Me avisa antes de que me pase.”
5. Consultar el resumen mensual y explicar qué muestra	Sí	Identificó todos los elementos correctamente y valoró la claridad: “Todo está en un solo lugar, no hay que buscar.”

Figura 20. Tabla Ana

Pruebas de usabilidad – Luis Fernández

Rol: Estudiante universitario (22 años)

Fecha: 29 de Octubre de 2025

Herramienta usada: Prototipo funcional en GitHub Pages

Tarea / Escenario	¿Completada?	Observaciones
1. Asignar un presupuesto mensual de \$1.000.000	Sí	Lo hizo en menos de 10 segundos. Dijo: *"Directo y sin rodeos."*
2. Registrar un gasto de \$300.000 con la descripción "Mercado"	Sí	No tuvo dudas. Comentó: "El campo de descripción es útil para llevar control."
3. Registrar un ingreso de \$200.000 con la descripción "Venta de ropa"	Sí	Entendió que los ingresos son informativos. Dijo: "Esto me ayuda a ver cuánto gané por fuera del presupuesto."
4. Registrar un gasto adicional de \$500.000 para alcanzar el 80% del presupuesto	Sí	Al ver la alerta naranja, exclamó: "¡Genial! Me avisa justo cuando voy por el 80%."
5. Consultar el resumen mensual y explicar qué muestra	Sí	Explicó con precisión cada valor y el propósito de la alerta. Añadió: "Es simple, pero cubre todo lo que necesito."

Figura 21. Tabla Luis

Pruebas de usabilidad – Sofía López

Rol: Madre cabeza de hogar (40 años)

Fecha: 29 de Octubre de 2025

Herramienta usada: Prototipo funcional en GitHub Pages

Tarea / Escenario	¿Completada?	Observaciones
1. Asignar un presupuesto mensual de \$1.000.000	Sí	Comentó: “Así empiezo con un límite claro, no gasto de más.”
2. Registrar un gasto de \$300.000 con la descripción “Mercado”	Sí	Llenó todos los campos sin ayuda. Dijo: “Me gusta que pida la fecha; así sé cuándo fue el gasto.”
3. Registrar un ingreso de \$200.000 con la descripción “Venta de ropa”	Sí	Reconoció que los ingresos no afectan el presupuesto. Comentó: “Los ingresos son míos, pero el presupuesto es solo para gastos.”
4. Registrar un gasto adicional de \$500.000 para alcanzar el 80% del presupuesto	Sí	Al ver la alerta, dijo: “Esto me da tranquilidad. Sé cuándo debo parar.”
5. Consultar el resumen mensual y explicar qué muestra	Sí	Explicó correctamente todos los elementos y valoró la alerta: “La naranja llama la atención; no se pasa por alto.”

Figura 22. Tabla Sofia

Patrones y arquitectura de software

El sistema se desarrolló bajo una arquitectura monolítica en el navegador, donde todo el código y la lógica residen en el cliente (navegador web). Dado el enfoque offline, la simplicidad del MVP y la necesidad de autonomía total, no se utiliza backend, servidor ni conexión a internet.

Patrones arquitectónicos y de diseño aplicados

- Arquitectura en capas (frontend-only):

Aunque el sistema es completamente local, se separaron responsabilidades en tres capas lógicas para mantener claridad y organización:

- Presentación: archivos HTML (index.html, registrar.html, resumen.html) que muestran la interfaz al usuario.
- Lógica del sistema: archivo script.js, que gestiona el cálculo del saldo, la activación de la alerta del 80% y la validación de entradas.
- Persistencia: localStorage, que almacena el presupuesto asignado y las transacciones registradas directamente en el navegador.
- Patrón de diseño: Controlador centralizado:
El archivo script.js actúa como un controlador único que:
 - Valida y guarda nuevas transacciones (gastos o ingresos).
 - Calcula el saldo actual como Presupuesto – Gastos acumulados.
 - Evalúa si los gastos han alcanzado el 80% del presupuesto para activar la alerta visual.
 - Lee y escribe datos en localStorage.
- Principios SOLID aplicados:
 - Responsabilidad Única: cada función en script.js tiene un propósito claro y único (ej. getSaldoActual(), estaEnAlerta()).
 - Abierto/Cerrado: es posible extender funcionalidades (ej. agregar exportación de datos) sin alterar el código existente.

Justificación de la arquitectura

Esta arquitectura fue seleccionada porque:

- Garantiza autonomía offline: no depende de servidores, bases de datos externas ni conexión a internet.

- Es simple y mantenible: ideal para un MVP desarrollado por una sola persona, con recursos limitados.
- Cumple con los requisitos no funcionales: usabilidad intuitiva, accesibilidad visual y tiempo de respuesta inmediato (sin esperas de red).
- Se alinea con la necesidad real del usuario: madres cabeza de hogar y estudiantes que requieren una herramienta confiable, sin complejidad técnica.

Diagrama de arquitectura del desarrollo

El sistema se compone de tres módulos integrados en el navegador:

- Interfaz de usuario: HTML/CSS
- Lógica del sistema: JavaScript (script.js)
- Persistencia: localStorage



Figura 23. Diagrama Arquitectura

Estructura del proyecto en GitHub

La organización del código refleja esta arquitectura:

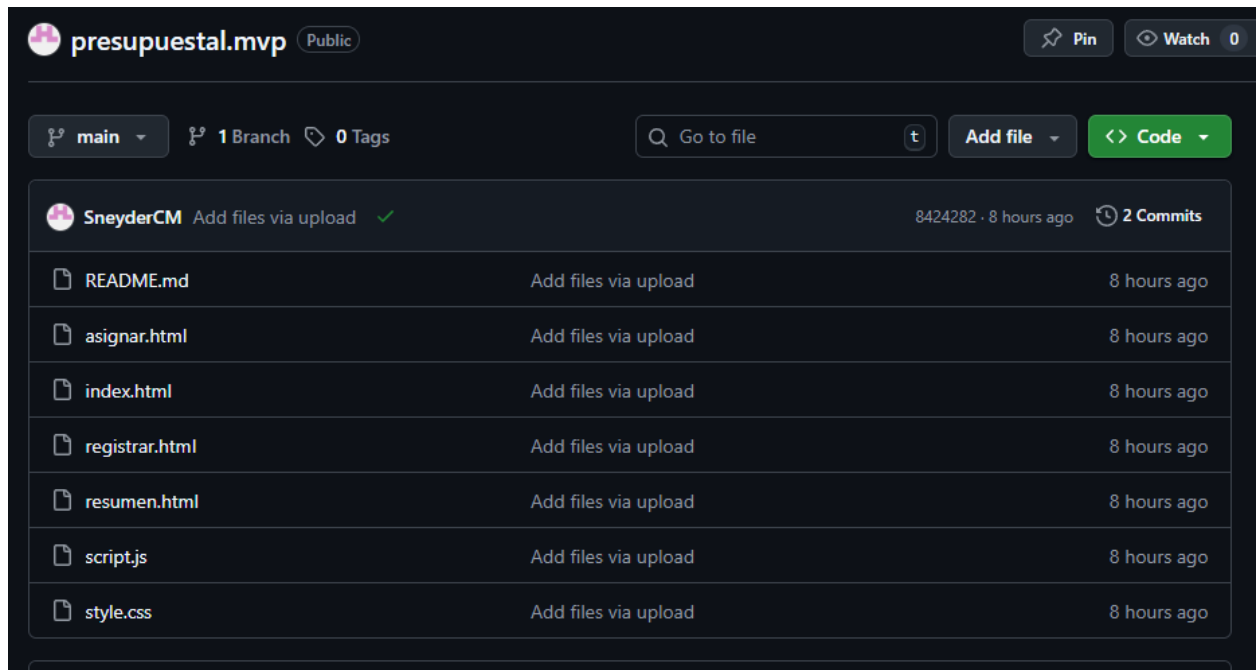


Figura 24. Estructura GitHub

Vídeo de demostración (elevator pitch)

Enlace del video: https://youtu.be/qNDVULKTbZg?si=2tcEwrH_01sUtzjA

GitHub: Repositorio entrega trabajos:

<https://github.com/SneyderCM/ProyectoDeSoftware.git>

Repositorio: presupuestal.mvp <https://github.com/SneyderCM/presupuestal.mvp.git>

Conclusión

Este proyecto demuestra que es posible desarrollar tecnología con propósito, centrada en quienes más la necesitan pero menos acceso tienen. Al formular, diseñar y preparar la implementación de una aplicación de escritorio autónoma para el control presupuestal, se responde a una necesidad real de madres cabeza de hogar en contextos de baja conectividad, sin exigirles más de lo que pueden dar. El enfoque en simplicidad, claridad y autonomía offline no solo es viable, sino urgente. Este trabajo sienta las bases para un MVP funcional que, en fases posteriores, podría convertirse en una herramienta real de empoderamiento financiero.

Bibliografía

- *Pressman, Roger S., (2021) Ingeniería de software. McGraw-Hill Interamericana. Capitulo 24, 25, 26 página de la 490 a 548*
- *Omaña, M. (2012). Manufactura esbelta: una contribución para el desarrollo de software con calidad. Red Enlace página de la 14 a 18.*
- *Sommerville, I. (2005). Ingeniería del software. Pearson educación.*
- *Universidad Iberoamericana. (2025). Actividad 1 – Identificar el proyecto tecnológico a trabajar*
- *Cardona, S. (2025). Act. 2 – Documento de formulación del proyecto [Trabajo de curso no publicado]. Proyecto de software, Comunidad Universitaria Iberoamericana.*