

UNIVERSIDAD MINUTO DE DIOS

Tecnología en informática – Proyecto Final

DOCUMENTACIÓN TÉCNICA DEL PROYECTO

FinAi – Plataforma Web de Educación Financiera

Presentado por:

Luis David Yate Pulido

Programa:

Tecnología en informática

Asignatura:

Transferencia de Conocimientos para el Desarrollo de Aplicaciones

Docente:

Ing. RUBEN DARIO GONZALEZ BARRERA

Fecha:

Bogotá D.C., noviembre de 2025

1. Documento de Requisitos del Sistema (DRS)

1.1 Introducción:

- **Propósito del documento:**

Este documento define los requisitos técnicos y de usuario del proyecto **FinAi**, una aplicación web educativa enfocada en mejorar el manejo de ingresos, gastos y hábitos financieros. La plataforma ofrece contenido guiado, herramientas prácticas, recordatorios de pago y explicaciones claras sobre conceptos financieros como cuotas, intereses y planificación de compras.

- **Alcance del proyecto:**

FinAi SÍ hace:

- Ofrecer módulos de educación financiera interactivos.
- Registrar ingresos y gastos localmente (LocalStorage).
- Mostrar balance financiero del usuario.
- Explicar conceptos financieros con ejemplos y simuladores.
- Permitir configurar recordatorios básicos.
- Funcionar de manera responsiva en web (PC y móvil).

FinAi NO hace:

- Realizar transacciones reales.
- Conectarse a entidades financieras.
- Solicitar datos bancarios.
- Requerir backend o base de datos en servidor (versión 1.0).

Definiciones:

- **Frontend:** Interfaz visible construida con HTML, CSS y JS.

- **LocalStorage:** Base de datos local del navegador.
- **UI/UX:** Interfaz de usuario / experiencia de usuario.
- **JS Vanilla:** JavaScript sin frameworks.

Referencias

- MDN Web Docs
- Material de documentación del proyecto FinAi
- W3C Standards

1.2 Descripción General

- **Perspectiva del producto:**

FinAi funciona como una aplicación web educativa auto-contenida. Toda la lógica se ejecuta en el navegador del usuario usando JavaScript. No necesita servidores ni APIs externas. Permite aprender conceptos financieros y gestionar un pequeño registro económico personal.

- **Funciones principales:**

- Registro de ingresos y gastos.
- Visualización de balance mensual.
- Módulos educativos sobre intereses, cuotas, ahorro, presupuesto, etc.
- Generación de recordatorios básicos.
- Recomendaciones basadas en movimientos guardados.

- **Características del usuario:**

- Personas de 15 a 40 años interesadas en aprender finanzas personales.
- Usuarios sin experiencia previa en herramientas financieras.

- Usuarios que usan dispositivos móviles o PC.

- **Restricciones técnicas:**

- Requiere navegador moderno (Chrome, Edge, Firefox).
- Requiere habilitar JavaScript.
- Requiere compatibilidad con LocalStorage.
- No funciona offline si no se descarga como PWA (versión futura).

- **Supuestos:**

- El usuario tiene acceso a internet.
- El usuario utiliza la aplicación de forma recurrente.

1.3 Requisitos Específicos:

Código	Descripción
RF-001	Registrar ingresos y gastos mediante formularios HTML
RF-002	Validar datos ingresados antes de guardarlos.
RF-003	Guardar información localmente con LocalStorage.
RF-004	Mostrar balance total dinámicamente.
RF-005	Mostrar módulos educativos en formato interactivo.
RF-006	Permitir configurar recordatorios (texto + fecha).
RF-007	Mostrar consejos según los movimientos registrados.
RF-008	Permitir eliminar o editar un movimiento.

Requisitos No Funcionales (RNF):

Código	Descripción
RNF-001	El sitio debe ser completamente responsive (mobile-first).
RNF-002	El tiempo de carga inicial debe ser < 1.5 segundos.
RNF-003	La información debe persistir correctamente usando LocalStorage.
RNF-004	La interfaz debe ser clara e intuitiva para usuarios desde 12 años.
RNF-005	El código debe estar organizado en módulos JS para mantenimiento.
RNF-006	El diseño debe seguir un estilo moderno y minimalista.

2. Casos de Uso

2.1 Actores:

- Usuario final

2.2 Casos de Uso Principales

- CU-001: Registrar un ingreso o gasto:

Actor: Usuario

Precondición: Estar en la sección "Movimientos".

Flujo principal:

1. El usuario abre el formulario.
2. Ingresá valor, categoría y tipo.

- 3. JS valida los campos.**
- 4. LocalStorage guarda la información.**
- 5. La interfaz se actualiza con el nuevo movimiento.**
- 6. Se recalcula el balance.**

Flujos alternativos:

- **Datos inválidos → mensaje de error.**
- **Cancelación → volver al menú principal.**

Postcondición: Información almacenada correctamente.

- **CU-002: Consultar balance**

Flujo principal:

- 1. El usuario ingresa a la sección “Balance”.**
- 2. JavaScript calcula ingresos, gastos y diferencia.**
- 3. Se muestra balance total de forma gráfica o numérica.**

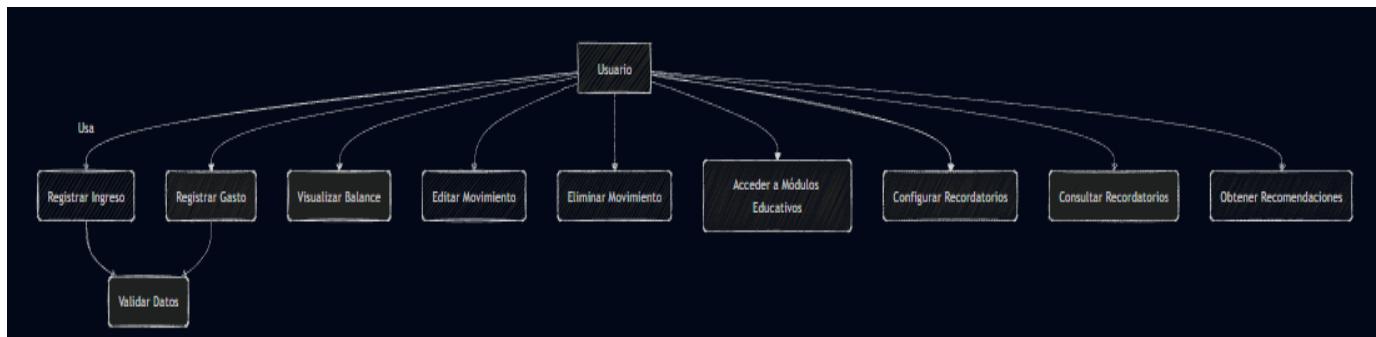
- **CU-003: Acceder a módulo educativo**

Flujo principal:

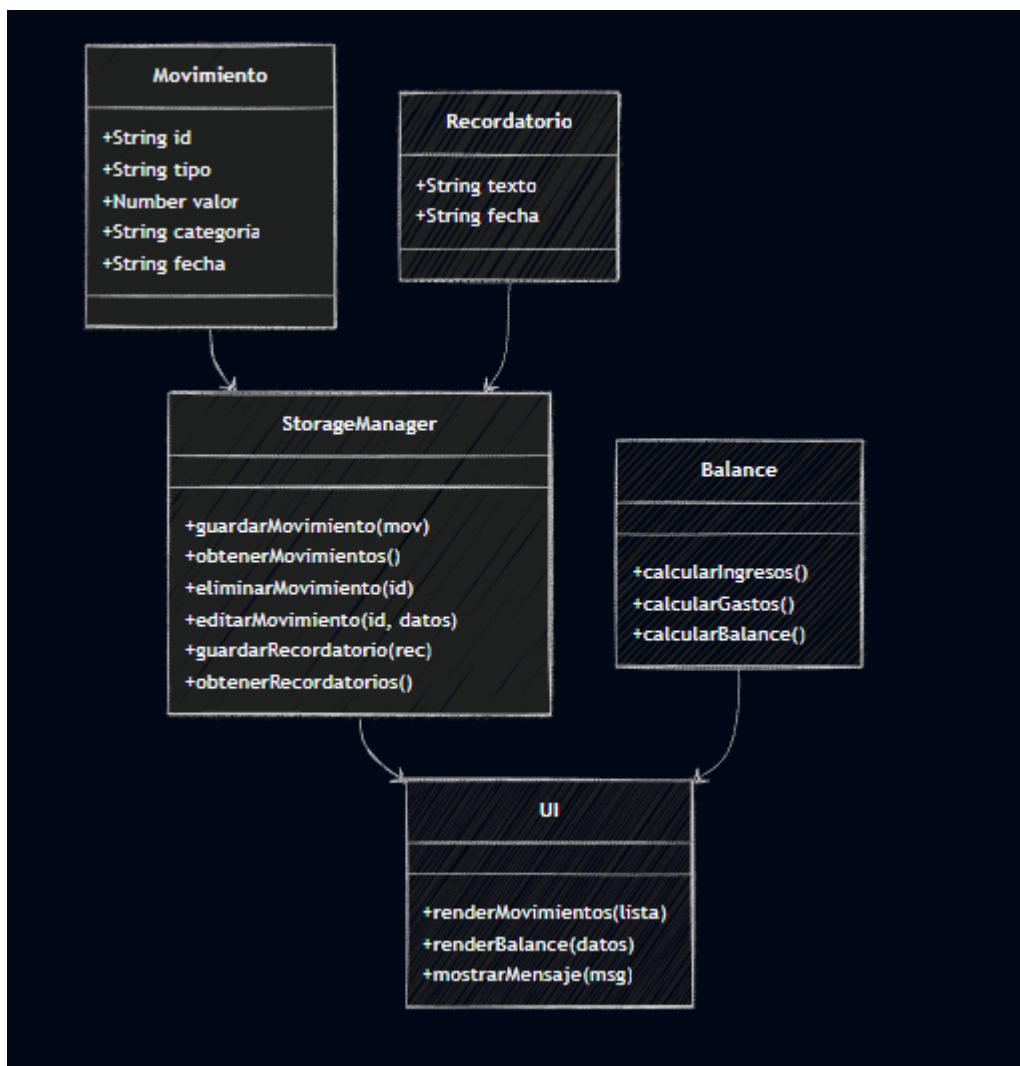
- 1. El usuario selecciona un módulo (ej. “Intereses”).**
- 2. Se carga contenido HTML+CSS+JS.**
- 3. El usuario interactúa con ejemplos o simuladores.**

3. Diagramas UML (adaptados a una app web):

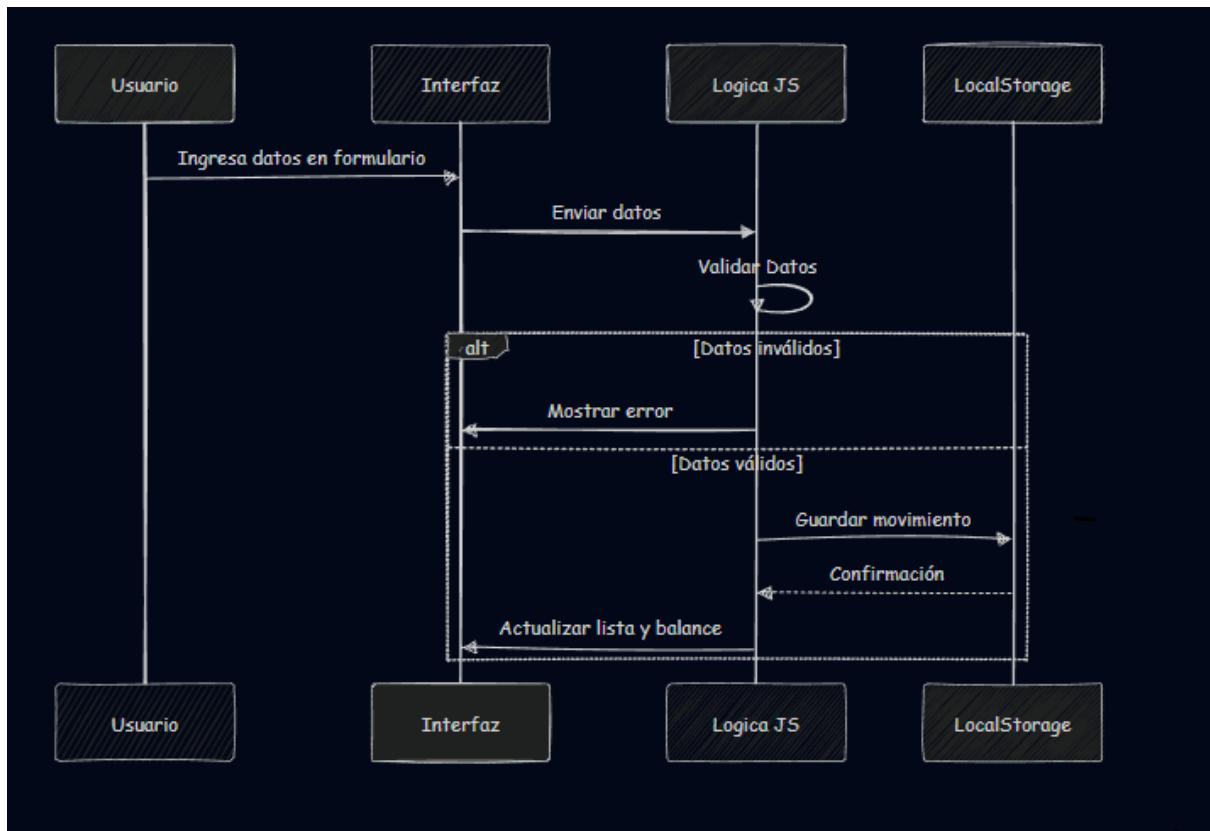
- Diagrama de caso de uso:



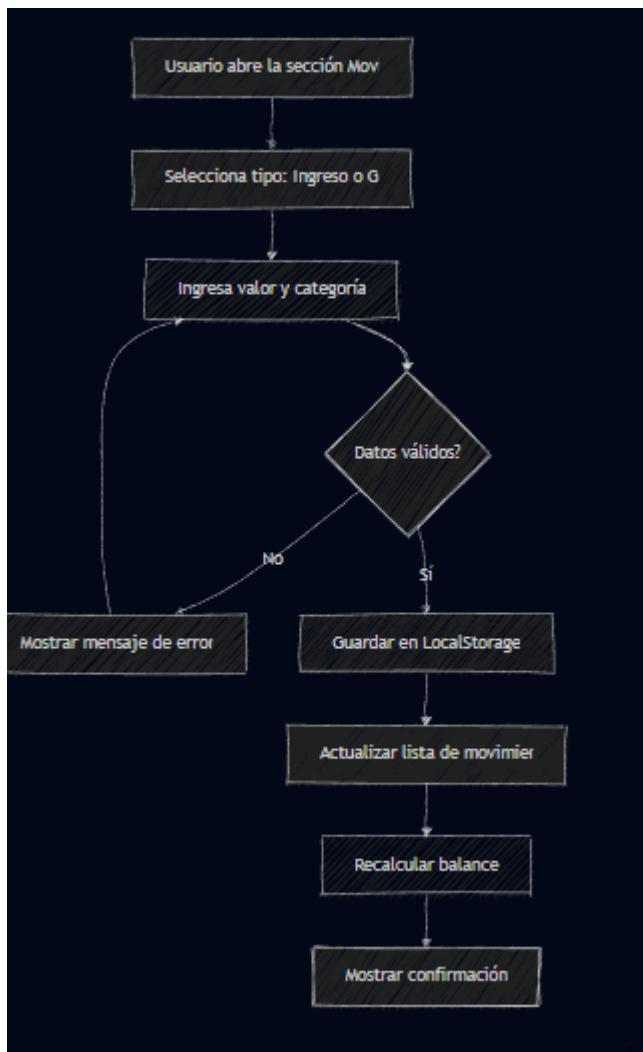
- Diagrama de clases:



- **Diagrama de secuencias:**



- **Diagrama de Actividades:**



4. Arquitectura de la Aplicación (HTML, CSS, JS)

FinAi está organizada siguiendo una arquitectura de frontend simple pero modular.

4.1 Estructura por Capas:

Capa de Presentación (HTML + CSS)

- Diseño responsive basado en Flexbox/Grid
- Paleta de colores suaves y profesionales
- Componentes visuales:

- **Formulario de movimientos**
- **Tablas / listas**
- **Sección educativa**
- **Menú superior y navegación lateral**

Capa de Lógica (JavaScript):

- **Validación de formularios**
- **Gestión de movimientos**
- **Cálculo del balance**
- **Renderizado dinámico de elementos**
- **Sistema de recordatorios**
- **Recomendaciones basadas en reglas simples**

Todo modularizado en archivos como:

```
/js
  movements.js
  reminders.js
  education.js
  balance.js
  storage.js
  ui.js
```

Capa de Datos

- **Uso de LocalStorage con claves como:**
 - "finai_user"
 - "finai_movimientos"

- o "finai_recordatorios"

4.2 Diccionario de Componentes (web):

Componente	Tipo	Funcion
index.html	HTML	Página principal y contenedor general
main.css	CSS	Estilos responsivos
app.js	Java Script	Lógica general
movimiento.js	Java Script	Gestión de ingresos/gastos
balance.js	Java Script	Cálculo del balance
education.js	Java Script	Contenido dinámico educativo
reminders.js	Java Script	Manejo de recordatorios
storage.js	Java Script	Gestión de LocalStorage

4.3 Base de Datos Local:

Se utilizará LocalStorage como base de datos local del navegador.

Estructura esperada:

```
{
  "finai_movimientos": [
    {
      "id": 1,
      "tipo": "ingreso",
      "valor": 500000,
      "categoria": "Salario",
      "fecha": "2025-02-24"
    }
  ],
}
```

```
"finai_recordatorios": [  
    {  
        "texto": "Pagar recibo de luz",  
        "fecha": "2025-01-05"  
    }  
]
```

Requisitos para utilizar la aplicación

Para usar FinAi correctamente necesitas:

- Un navegador moderno (Google Chrome, Edge, Firefox u Opera).
- Conexión a internet.
- Permitir la ejecución de JavaScript.
- Un dispositivo móvil o computador.

No requiere instalación ni registro.

3. Ingreso a la aplicación:

1. Abre tu navegador.
2. Ingresa a la dirección donde está publicada la aplicación (o abre el archivo `index.html` si estás en local).
3. Espera a que cargue la página principal.

La interfaz es intuitiva y está dividida por secciones visibles desde el menú.

4. Navegación principal:

FinAi cuenta con cuatro secciones principales:

4.1 Movimientos

Aquí puedes registrar tus ingresos y gastos diarios.

4.2 Balance

En esta sección podrás ver un resumen de tu situación financiera.

4.3 Educación financiera

Incluye módulos educativos sobre temas como intereses, cuotas, ahorro y presupuesto.

4.4 Recordatorios

Permite guardar recordatorios de pagos importantes.

Puedes navegar entre estas secciones mediante el menú principal.

5. Registrar ingresos o gastos

5.1 Cómo registrar un movimiento

1. En el menú selecciona Movimientos.
2. Elige si deseas agregar un Ingreso o un Gasto.
3. Completa los campos solicitados:
 - Valor
 - Categoría
 - Fecha
4. Haz clic en Guardar.

La aplicación guardará tu movimiento automáticamente.

5.2 Si aparece un error

Puede deberse a:

- Campos vacíos.
- Valor no numérico.
- Formato incorrecto.

Corrige los datos y vuelve a guardarlos.

6. Visualizar tu balance financiero

1. En el menú elige Balance.
2. La app mostrará:
 - Total de ingresos.
 - Total de gastos.
 - Balance final (positivo o negativo).

La información se actualiza automáticamente cada vez que registras un movimiento.

7. Editar o eliminar un movimiento

Editar un movimiento

1. En la tabla de movimientos busca el registro a editar.
2. Selecciona el botón Editar.
3. Modifica el valor o categoría.
4. Guarda los cambios.

Eliminar un movimiento

1. Ubica el movimiento que deseas borrar.
2. Haz clic en Eliminar.
3. Confirma la acción.

El balance se recalculará automáticamente.

8. Usar los módulos de Educación Financiera

1. En el menú selecciona Educación financiera.
2. Elige un tema:
 - Intereses
 - Ahorro
 - Cuotas
 - Presupuesto
3. Lee la información y utiliza los ejercicios interactivos para comprender mejor cada concepto.

9. Crear y consultar recordatorios

Para crear un recordatorio:

1. Ingresa en Recordatorios.
2. Escribe el título del recordatorio (ejemplo: “Pagar luz”).
3. Selecciona la fecha.
4. Haz clic en Guardar.

Para consultar tus recordatorios:

Simplemente ingresa a la misma sección y verás la lista completa.

10. Eliminar todos los datos

Si deseas borrar toda tu información:

1. Ingresa a la sección Configuración (si está disponible en tu versión).
2. Selecciona Limpiar datos.
3. Se eliminarán todos tus registros, recordatorios y balance.

11. Recomendaciones de uso

- Registra tus movimientos a diario para tener información precisa.
- Consulta el balance al menos una vez por semana.
- Revisa los módulos educativos para mejorar tu manejo financiero.
- Mantén actualizada la sección de recordatorios para evitar pagos atrasados.