# Plan de implementación y puesta en marcha

**Estado actual del proyecto al día de la fecha 23 de septiembre de 2025.**

Ya se encuentra implementado y probado en entorno local:

* **Gestión de movimientos**
  + RF1: Ingresar movimiento.
  + RF2: Detallar monto, tipo, categoría, fecha y descripción.
  + RF5: Editar movimiento.
  + RF6: Eliminar movimiento.
  + RF7: Listar movimientos en orden cronológico.
  + RF13: Actualización automática de la lista de movimientos.
* **Gestión de categorías**
  + RF3: Crear categoría.
* **Dashboard básico**
  + RF9: Mostrar ingresos, gastos y balance mensual.
* **Gestión de usuarios (autenticación)**
  + RF16: Crear cuenta.
  + RF17: Iniciar sesión.
  + RF18: Validar email único.
* **Presupuestos**
  + RF10: Crear presupuestos y notificar excedentes.

**Recursos necesarios para llevar a cabo la implementación del proyecto:**

**Técnicos:**

* **Backend:** 
  + Node.js: entorno de ejecución de JavaScript en el servidor.
  + Express.js: Framework para crear API REST y gestionar rutas.
  + PostgreSQL : base de datos.
  + Bcryp: encriptación de contraseñas.
  + JWT: autenticación y autorización mediante tokens.
  + Dotenv: manejo seguro de variables de entorno.
* **Frontend:** 
  + React: biblioteca para la construcción de la UI.
  + Vite
  + React Router: para la navegación entre paginas y protección de rutas privadas.
  + Fecth API: comunicación con la API backend desde el frontend.
* **Infraestructura:** GitHub (repositorio), Render (backend/DB), Vercel (frontend).
* **Colaboración:**, StarUML (diagramas), Postman (API).
* **Servicios externos:** sendgrid (utilizado para el envio de emails)

**Iteraciones del proyecto:**

## 5 iteración (Realizada) :

Fecha de inicio: 23 de agosto del 2025

Fecha final 23 de septiembre de 2025

Duración estimada: 4 semanas

Objetivo:

Tener una primera versión **completamente funcional** de la aplicación, con la **base de datos diseñada e implementada**, backend operativo y frontend conectado, además de los **requisitos funcionales centrales** ya listos y probados.

**Requisitos funcionales cubiertos**

* **RF1:** Ingresar un nuevo movimiento.
* **RF2:** Detallar monto, tipo (ingreso/gasto), categoría, fecha y descripción.
* **RF3:** Crear nueva categoría.
* **RF5:** Editar un movimiento.
* **RF6:** Eliminar un movimiento.
* **RF7:** Listar movimientos en orden cronológico.
* **RF9:** Mostrar total de ingresos, gastos y balance mensual.
* **RF10:** Crear presupuestos mensuales y notificar sobre su estado.
* **RF13:** Actualización automática de lista de movimientos.
* **RF16:** Registro de usuario (email + contraseña).
* **RF17:** Inicio de sesión de usuario.
* **RF18:** Verificación de emails únicos.

**Requisitos no funcionales cubiertos:**

* RNF1: EL Sistema debe estar protegido a accesos no autorizados a rutas privadas.

**Tareas realizadas:**

**Base de datos:**

Creación de tablas de usuarios, categorías, movimientos y presupuestos.

**Backend** (Express + PostgreSQL):

Se implementarios los endpoints REST principales.

Middleware de autenticación JWT activo en rutas protegidas.

**Testing:**

Realizar las pruebas unitarias para testear los requerimientos funcionales implementados.

**Documentación:**

* Descripción de casos de uso.
* Diagramas BPMN de procesos claves.
* Diagramas de secuencia de casos de uso.
* Diagrama de casos de usos.
* Descripción de endpoints y procesos.

**Entregables:**

* Código fuente backend y frontend.
* Base de datos implementada y conectada.
* Conjunto de pruebas unitarias.
* Aplicación funcional en entorno local.
* Documentación completa.

## 6 Iteración:

Duración: 4 semanas

Fecha de inicio 24 de septiembre.

Fecha finalización: 24 de octubre

**Objetivo:**

Agregar al sistema funcionalidades avanzadas que mejoren la experiencia del usuario y la seguridad de la aplicación, incluyendo:

* Visualización de gráficos.
* Filtrado de movimientos.
* Notificaciones de presupuesto.
* Seguridad mejorada (validación de contraseñas, verificación de email, protección de rutas).
* Documentación completa de nuevas funcionalidades.

**Requisitos funcionales cubiertos**

* **RF8:** Filtrar la lista de movimientos por mes, tipo y categoría.
* **RF11a:** Gráfico de barras que muestre total de ingresos y gastos.
* **RF11b:** Gráfico de torta con distribución de gastos por categoría.
* **RF12:** Gráfico de línea que represente el balance mensual a lo largo del tiempo.
* **RF14:** Guardado automático de todos los cambios y registros en la base de datos.
* **RF15:** Restablecer gastos e ingresos al iniciar un nuevo mes.
* **RF19:** Permitir al usuario cambiar la contraseña si se la olvida.

**Requisitos no Funcionales cubiertos:**

* **RNF2:** El sistema debe adaptar su contenido para mostrarse con claridad en dispositivos móviles.
* **RNF3:** Integración con SendGrid para envío de correos (verificación de email y recuperación de contraseña).

**Tareas a realizar:**

**Frontend:**

* Implementar gráficos con Recharts.
* Filtros interactivos para movimientos.
* Notificaciones visuales al superar presupuesto.
* Formularios para cambio de contraseña y verificación de email.

**Backend:**

* Validación y sanitización de entradas usando librerías como validator.js.
* Lista de revocación de tokens y control de sesiones.

**Testing:**

* Unitarios para endpoints de estadísticas y seguridad.
* Integración frontend-backend para gráficos y filtros.
* Pruebas de envío de correos y validación de tokens.
* Realizar pruebas de usabilidad en móvil y escritorio.
* Pruebas de seguridad (ataques de fuerza bruta,revocación de tokens, sanitación de tokens)

**Documentación:**

* Realización de diagrama entidad relación de la base de datos.

**Entregables:**

* Código fuente actualizado backend y frontend.
* Base de datos con nuevas tablas o campos para recuperación de contraseña y revocación de tokens.
* Tests unitarios e integración completos.
* Aplicación funcional con gráficos, filtros y seguridad mejorada.

7 iteración (Refinamiento y despliegue)

Duración estimada=4 semanas

Fecha inicio: 25 de octubre

Fecha final: 25 de noviembre

**Objetivo:** refinar la interfaz de usuario y realizar el despliegue de la aplicación en producción.

**Requisitos funcionales cubiertos**

No se agregan nuevos RF, se mejora lo existente.

Mejoras en RF ya implementados:

* Refinamiento del dashboard y gráficos.
* Ajuste de notificaciones de presupuesto.

**Tareas a realizar :**

* Realizar el refinamiento del frontend.
* Deploy y configuración:
* Backend en **Render** con variables de entorno seguras.
* Frontend en **Vercel** configurado con dominio.
* Configuración de CORS y HTTPS.
* Migración de la base de datos PostgreSQL local a la nube (Render)

Documentación final:

Manual técnico: es una recopilación organizada de toda la documentación generada durante el proyecto, el cual incluye:

* Arquitectura del sistema: diagrama de arquitectura y descripción de los componentes.
* Base de datos: Modelo entidad-relación, configuración de la BD en local y Render.
* Backend: estructura de carpetas del proyecto, dependencias principales, descripción de endpoints, configuración de variables de entorno.
* Frontend: Estructura de carpetas del proyecto, principales componentes, contextos usados y manejo de rutas protegidas.
* Testing: estrategia de pruebas (unitarias).
* Despliegue: Desccripcion de pasos para correr en local, y despliegue en Versel y Render.
* Mantenimiento y futuras mejoras: Cómo agregar nuevos endpoints, cómo extender el modelo de datos, requisitos no funcionales a mejorar (seguridad, performance, escalabilidad).

**Testing:**

* Pruebas end-to-end en los principales flujos: login, registrar movimiento, editar movimiento, filtrar, dashboard.
* Pruebas unitarias y de integración completadas en todas las capas

**Entregables:**

* Aplicación desplegada en Render (backend) y Vercel (frontend).
* Base de datos en la nube.
* Documentación final (manual técnico + usuario + plan de mantenimiento).
* Resultados del testing.