

**Proyecto de Integración Curricular
Sistema de Gestión de Finanzas Personales**



Leonardo Sebastián Agila Llumixi
Universidad Internacional del Ecuador
Facultad de Ciencias Técnicas
Ingeniería en Sistemas de información
PROGRAMACION ORIENTADA A OBJETOS
II-UIO-2025 SEMESTRE SEPT 2025 - EN 2026
ING. IVAN GALO REYES CHACON

Contenido

Introducción	3
Objetivo del Sistema.....	3
Alcance del Proyecto	3
Arquitectura del Sistema	4
Funcionalidades del Sistema	5
Extras y Mejoras Propuestas	5
Módulos Propuestos.....	6
Despliegue y modo "Área limitada"	8
Conclusión	9

Introducción

El presente documento describe la planeación y diseño de un Sistema de Gestión de Finanzas personales. Este sistema busca ofrecer a los usuarios una herramienta integral para manejar sus ingresos, gastos, deudas, suscripciones y presupuestos, con el objetivo de fomentar un mayor control y organización financiera. El proyecto será desarrollado en el lenguaje Go, bajo un enfoque modular y escalable, con soporte para API REST.

Objetivo del Sistema

El objetivo principal del sistema es permitir que los usuarios gestionen sus finanzas personales de manera eficiente, incorporando funciones avanzadas como control de presupuestos, seguimiento de metas de ahorro, generación de reportes y soporte para múltiples cuentas y monedas. Adicionalmente, el sistema incluirá un módulo de API para su integración con aplicaciones web o móviles, y la posibilidad de ser usado por una comunidad pequeña en un área geográfica definida.

Alcance del Proyecto

El sistema cubrirá las siguientes áreas clave:

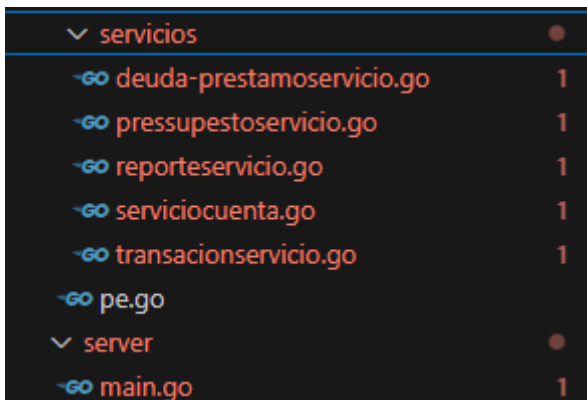
- Gestión de cuentas múltiples (bancarias, efectivo, tarjetas).
- Registro de ingresos y egresos.
- Gestión de presupuestos mensuales por categoría.
- Control de deudas y préstamos.
- Registro de suscripciones recurrentes.
- Exportación e importación de datos.
- Reportes detallados en PDF/CSV.
- Manejo de múltiples monedas con tasas actualizadas.
- Módulo de API REST en Go.
- Soporte para pequeños grupos de usuarios en áreas locales.

Arquitectura del Sistema

El sistema estará basado en una arquitectura cliente-servidor con un backend en Go. La API REST permitirá gestionar todas las funcionalidades, mientras que los datos se almacenarán en una base de datos relacional como PostgreSQL. La aplicación podrá ser accedida a través de un frontend web o una aplicación móvil. Arquitectura propuesta:

[Usuario] → [Frontend Web/Móvil] → [API REST en Go] → [Base de Datos PostgreSQL]

▼ proyectofinal	●
▼ api	●
▼ controladores	●
-Go controlador-cuenta.go	1
-Go controlador-debito.go	1
-Go controlador-presupuesto.go	1
-Go controlador-usuario.go	1
-Go transacion.cuenta.go	1
-Go rutas.go	1
▼ logica	●
▼ entidades	●
-Go cuenta.go	1
-Go deuda-prestamo.go	1
-Go presupuesto.go	1
-Go transacciones.go	1
-Go usuario.go	1
▼ repositorios(bd)	●
-Go cuentarepo.go	1
-Go presupestorepo.go	1
-Go reporterepo.go	1
-Go transacionrepo.go	1
▼ servicios	●
-Go deuda-prestamoservicio.go	1
-Go pressupestoservicio.go	1
-Go reporteservicio.go	1



▼ servicios	●
➔ deuda-prestamoservicio.go	1
➔ presupuestoservicio.go	1
➔ reporteservicio.go	1
➔ serviciocuenta.go	1
➔ transacionservicio.go	1
➔ pe.go	
▼ server	●
➔ main.go	1

Funcionalidades del Sistema

- Multicuenta y transferencias internas.
- Gestión de presupuestos y alertas.
- Calendario de gastos e ingresos recurrentes.
- Exportación e importación de datos (CSV, PDF).
- Reportes avanzados de flujo de caja.
- Soporte para múltiples monedas.
- Control de deudas y préstamos.
- Sistema multiusuario con cuentas compartidas.
- Seguridad (2FA, encriptación de datos).
- Respaldo automático de la base de datos.
- Dashboard interactivo con resúmenes gráficos.

Extras y Mejoras Propuestas

- Módulo de metas de ahorro con visualización de progreso.

- Control de suscripciones recurrentes.
- Posibilidad de manejar finanzas personales y de pequeños negocios.
- Bitácora de auditoría de acciones de usuario.
- Módulo offline con sincronización posterior.
- API REST que permitirá integrar el sistema con aplicaciones externas.
- Escalabilidad para comunidades pequeñas en áreas locales.

Módulos Propuestos

Módulo A — Autenticación y seguridad

- Registro/login (email + contraseña).
- OAuth2 (Google) opcional.
- JWT para autenticación de API.
- TOTP (2FA) o verificación por correo.
- Roles: admin, usuario.

Módulo B — Cuentas y monedas

- Crear/editar/eliminar cuentas (banco, tarjeta, efectivo).
- Soporte multi-moneda por cuenta.
- Conversión de moneda mediante llamada a API de tasas (opcional).

Módulo C — Transacciones

- Registrar ingresos, gastos y transferencias internas.
- Asociar categoría, etiqueta, cuenta y nota.
- Marcar transacciones como recurrentes (sistema de recurrencia configurable).

Módulo D — Presupuestos

- Crear presupuestos mensuales por categoría.
- Seguimiento y alertas (umbral configurable).

Módulo E — Metas de ahorro

- Crear metas (monto, fecha límite).
- Simulación de plan de ahorro (sin IA: cálculo matemático determinista).
- Visualización de progreso.

Módulo F — Suscripciones y pagos recurrentes

- Registrar suscripciones (monto, frecuencia, próximo pago).
- Cálculo de gasto anual por suscripciones.

Módulo G — Deudas y préstamos

- Registrar préstamos recibidos/otorgados.
- Registrar pagos parciales y calcular saldo.

Módulo H — Calendarización y recordatorios

- Calendario con eventos financieros (vencimientos, sueldos).
- Notificaciones por email o push (cuando aplique).

Módulo I — Import/Export y reportes

- Importar CSV/Excel bancario (mapeo de columnas configurable).
- Exportar transacciones a CSV, Excel y PDF.
- Reportes: flujo de caja, por categorías, comparativo mensual/anual.

Módulo J — OCR de facturas (sin IA)

- Subida de imagen/PDF, extracción con Tesseract OCR para campos: fecha, monto, NIT/cédula, proveedor.
- Asistente de validación manual (usuario confirma/corrigir los datos antes de guardar).

Módulo K — Generación de documentos

- Generar facturas o comprobantes en PDF con gofpdf.
- Opción: incluir código QR con link a comprobante/verificación.

Módulo L — Auditoría y backups

- Registro de cambios (usuario, timestamp, acción).
- Exportar backup en JSON/SQL.
- Restauración básica desde backup.

Módulo M — Modo offline/Sync (para área limitada)

- Frontend almacena localmente (IndexedDB / SQLite móvil).
- Sincronización con la API cuando hay conectividad.
- Resolución de conflictos simple: última modificación gana o estrategia por usuario/admin.

Despliegue y modo "Área limitada"

Se plantea una opción de despliegue local / área limitada pensada para un público pequeño (ej. vecindario, ONG local, pyme con varias sucursales en una ciudad pequeña):

- Opción A — Servidor local (por ejemplo Raspberry Pi o servidor local)

Instalar Docker + Docker Compose.

Servicios: API Go + PostgreSQL + Redis (opcional) + Nginx para HTTPS.

Ventajas: datos controlados físicamente, latencia baja en red local.

Requisitos: servidor con al menos 4GB RAM (mejor 8GB), SSD, conexión de red local estable.

- Opción B — Red local con réplica por sucursal

Nodo central (base de datos principal) y nodos locales que sincronizan con el central.

Mecanismo: replicación asincrónica o export/import periódicos.

Útil para lugares con conectividad intermitente.

- Opción C — Instancia en mini-datacenter local (ISP local)

Contratar VPS local con disponibilidad en la ciudad.

Permite HTTPS con certificados locales y control de acceso geográfico.

- Consideraciones para área limitada

Autenticación reforzada: limitar intentos de login por IP de la red local.

Política de privacidad: acuerdos de uso adaptados al contexto local.

Soporte offline: modos de operación cuando la conexión al servidor central falla (trabajo local y sincronización posterior).

Escalabilidad: empezar con una sola instancia Docker Compose; si crece, migrar a Kubernetes local (k3s).

Conclusión

El Sistema de Gestión de Finanzas Personales propuesto representa una solución ambiciosa y práctica para el control de finanzas de individuos o pequeños grupos. Gracias a su diseño modular, soporte multiusuario, múltiples funciones avanzadas y la inclusión de una API REST, el sistema tiene el potencial de evolucionar hacia una plataforma escalable, adaptable y de gran utilidad para distintos contextos.

