

# Integral Finanzas – Requisitos funcionales (Web + iOS + Android)

**Documento:** Benchmark → Catálogo de requisitos (tipo PRD + backlog) **Fecha:** 2026-02-24 **Estado:** Borrador para implementación rápida (MVP) + roadmap por versiones

## 0) Objetivo del producto

Construir una aplicación **muy completa** de finanzas personales (Web + App móvil) que combine:

- Registro y control financiero (manual + importación + opcionalmente sincronización bancaria)
- Presupuesto avanzado (múltiples métodos)
- Suscripciones y facturas (calendario y alertas)
- Patrimonio (net worth), metas, deuda, inversiones
- Reportes/insights y colaboración por hogar

### 0.1 Productos benchmark (origen de funcionalidades)

Base solicitada: Bluecoins + Quicken Simplifi Otras líderes: YNAB, Monarch Money, Rocket Money, Empower Personal Dashboard, NerdWallet

Fuentes primarias usadas (oficiales):

- Google Play listings: Bluecoins, Quicken Simplifi, Rocket Money, Empower, etc. (URLs en Anexo A)
- Centros de ayuda / páginas oficiales: Quicken Simplifi (Spending Plan y Projected Cash Flow), Monarch (Recurring Calendar), NerdWallet (Subscriptions & Bills), Rocket Money (Pricing y Manage Subscriptions), YNAB (Pricing)

## 1) Alcance por versiones (sin re-trabajo)

La idea es que cada versión sea un **subset estable**. No se reescribe el documento; solo se implementa en orden.

### 1.1 MVP (para salir rápido a producción)

**Meta:** operación estable de registro + presupuesto simple + recurrentes + reportes básicos.

- Onboarding (manual + import CSV)
- Cuentas + transacciones + categorías + split
- Presupuesto (modo “Spending Plan” simplificado)
- Recurrentes: recordatorios + calendario básico
- Net Worth básico
- Reportes base

- Seguridad mínima (PIN/biometría en móvil, cifrado en tránsito, roles básicos si hay hogar)

### **1.2 V1.1 – V1.3 (eleva el valor percibido)**

- Reglas automáticas (categorización, tags, asignación)
- Detección de recurrentes (auto) + centro de suscripciones
- Metas + contribuciones + forecast 6–12 meses
- Multi-monedas + tasas

### **1.3 V2 (diferenciadores fuertes)**

- Hogares/colaboración avanzada + auditoría por miembro
- Proyección cash-flow tipo “Projected Cash Flow” por cuenta
- Deuda: amortización + estrategias (bola de nieve / avalancha)
- Inversiones: holdings + rendimiento
- Motor de insights (anomalías, variaciones, alertas inteligentes)

### **1.4 V3 (extensiones opcionales según país/mercado)**

- Integración bancaria (si hay agregador confiable)
- Cancelación asistida de suscripciones (concierge) / negociación de facturas (si aplica legal y comercialmente)
- Marketplace/afiliación (tarjetas, préstamos) y educación financiera

## **2) Arquitectura funcional (módulos)**

1. Identidad, hogares y permisos  
 2. Onboarding e ingestión de datos (manual/import/sync opcional)  
 3. Cuentas y monedas  
 4. Transacciones (incl. split, tags, adjuntos)  
 5. Categorías y merchants  
 6. Reglas y revisión (“needs review”)  
 7. Presupuesto (3 métodos)  
 8. Recurrentes, bills y suscripciones (calendario)  
 9. Cash Flow / Forecast (proyección)  
 10. Metas y planificación (ahorro)  
 11. Deuda (préstamos/tarjetas)  
 12. Patrimonio (net worth) e inversiones  
 13. Reportes e insights  
 14. Exportación/backup/auditoría  
 15. Notificaciones  
 16. Seguridad y cumplimiento (no-funcional)  
 17. Administración/operación (observabilidad)

## **3) Modelo de dominio (entidades y campos mínimos)**

> Esto define las tablas/colecciones lógicas. Puedes mapearlo a SQL o NoSQL.

### **3.1 Identidad y colaboración**

#### **User**

- user\_id (UUID), email/phone, nombre, locale, tz, created\_at, last\_login\_at, status

#### **Household**

- household\_id, nombre, owner\_user\_id, moneda\_base, created\_at, plan\_tier, status

## **HouseholdMember**

- household\_id, user\_id, role (OWNER|ADMIN|MEMBER|VIEWER), joined\_at, status

## **PermissionOverride (opcional V2)**

- subject (account\_id/category\_id/report\_id), member\_user\_id, permission (READ|WRITE|ADMIN)

## **3.2 Cuentas y monedas**

### **Account**

- account\_id, household\_id, type (CASH|CHECKING|SAVINGS|CREDIT\_CARD|LOAN|INVESTMENT|PROPERTY|OTHER)
- nombre, institución (texto), currency\_code, opening\_balance, current\_balance (derivado u opcional), status
- include\_in\_budget (bool), include\_in\_net\_worth (bool), archived\_at

### **ExchangeRate**

- date, from\_currency, to\_currency, rate, source (manual|provider), created\_at

## **3.3 Categorías, tags y merchants**

### **Category**

- category\_id, household\_id, parent\_category\_id (nullable), nombre, type (INCOME|EXPENSE|TRANSFER\_EXCLUDE), status

### **Tag**

- tag\_id, household\_id, nombre, color (opcional), status

### **Merchant**

- merchant\_id, household\_id (o global), display\_name, normalized\_name, mcc (opcional), default\_category\_id (opcional)

## **3.4 Transacciones**

### **Transaction**

- txn\_id, household\_id, account\_id
- txn\_date (fecha contable), posted\_date (fecha real del banco/import), amount, currency\_code
- direction (INFLOW|OUTFLOW|TRANSFER)
- merchant\_id (nullable), payee\_text, category\_id (nullable), notes, status (PENDING|CLEARED|RECONCILED|VOID)
- is\_recurring\_match (bool), recurring\_instance\_id (nullable)
- created\_by, created\_at, updated\_at, source (manual|import|sync), external\_id (para dedupe)

### **TransactionSplit**

- split\_id, txn\_id, category\_id, amount, notes (opcional)

### **TransactionAttachment**

- attachment\_id, txn\_id, file\_ref, mime\_type, created\_at

### **TransactionTag**

- txn\_id, tag\_id

## **3.5 Reglas y revisión**

### **Rule**

- rule\_id, household\_id
- priority (int), enabled (bool)
- match: merchant/payee contains>equals, amount range, account\_id, direction, keywords, day\_of\_month, etc.
- actions: set\_category, set\_merchant, add\_tag(s), set\_needs\_review, auto\_split\_template\_id, assign\_to\_member\_id

### **ReviewQueueItem**

- item\_id, txn\_id, reason (UNCATEGORIZED|SPLIT\_SUSPECT|MERCHANT\_UNKNOWN|RULE\_CONFLICT|USER\_FLAG), assigned\_to\_user\_id, status (OPEN|DONE), created\_at

## **3.6 Presupuestos**

### **BudgetPeriod**

- budget\_period\_id, household\_id, period\_start, period\_end, method (CATEGORY|FLEX|ENVELOPE|SPENDING\_PLAN), status (OPEN|LOCKED)

### **BudgetLine (CATEGORY method)**

- line\_id, budget\_period\_id, category\_id, planned\_amount, rollover\_enabled, rollover\_from\_prev (amount), alerts\_threshold\_pct

### **FlexBucket (FLEX method)**

- bucket\_id, budget\_period\_id, name, planned\_amount, included\_categories (list)

### **EnvelopeAllocation (ENVELOPE method)**

- alloc\_id, budget\_period\_id, category\_id, assigned\_amount, available\_amount, target\_type (MONTHLY|BY\_DATE|BALANCE), target\_value

### **SpendingPlanItem (SPENDING\_PLAN method)**

- item\_id, household\_id, type (INCOME|BILL|SUBSCRIPTION|TRANSFER|SAVINGS\_GOAL|PLANNED\_SPEND|OTHER\_SPEND)
- name, account\_id (optional), category\_id (optional), amount\_rule (FIXED|AVERAGE\_LAST\_N), amount\_value, schedule\_id (nullable), active

## 3.7 Recurrentes, bills y suscripciones

### Schedule

- schedule\_id, household\_id
- frequency (DAILY|WEEKLY|BIWEEKLY|MONTHLY|YEARLY|CUSTOM)
- by\_day\_of\_week (list), by\_day\_of\_month (int), interval (int), start\_date, end\_date (nullable), timezone
- next\_run\_at (computed), status

### RecurringSeries

- series\_id, household\_id, kind (BILL|INCOME|SUBSCRIPTION|TRANSFER|PAYCHECK|OTHER)
- name, merchant\_id (nullable), category\_id (nullable), account\_id (nullable), amount\_rule, amount\_value
- schedule\_id, detection\_confidence (0..1), source (manual|detected), active

### RecurringInstance

- instance\_id, series\_id, due\_date, expected\_amount, currency\_code
- state (EXPECTED|MATCHED|PAID|SKIPPED|RESCHEDULED|CANCELED|OVERDUE)
- matched\_txn\_id (nullable), resolved\_by\_user\_id (nullable), resolved\_at (nullable)

### Subscription (vista especializada sobre recurring)

- subscription\_id, household\_id, series\_id, provider\_name, billing\_cycle, last\_charge\_date, next\_charge\_date, status (ACTIVE|CANCELED|PAUSED), cancel\_method (manual|assisted), notes

## 3.8 Metas, deuda, inversiones

### Goal

- goal\_id, household\_id, name, type (SAVING|DEBT\_PAYDOWN|EMERGENCY|CUSTOM)
- target\_amount, target\_date (nullable), linked\_account\_id (nullable), status

### GoalContribution

- contrib\_id, goal\_id, date, amount, source\_txn\_id (nullable)

### DebtPlan

- plan\_id, household\_id, strategy (avalanche|snowball|custom)
- included\_accounts (loan/cc list), extra\_payment\_amount, start\_date, projections

### **Holding / InvestmentPrice (V2)**

- holding\_id, account\_id, symbol, name, quantity, avg\_cost, currency\_code
- price\_date, price, source

### **NetWorthSnapshot**

- snapshot\_id, household\_id, date, total\_assets, total\_liabilities, net\_worth, breakdown\_json

## **3.9 Reportes y exportación**

### **ReportDefinition**

- report\_id, household\_id, name, type  
(SPEND\_BY\_CATEGORY|CASHFLOW|NETWORTH|SUBSCRIPTIONS|CUSTOM)
- filters\_json, chart\_pref, created\_by, created\_at

### **ExportJob**

- export\_id, household\_id, format (CSV|QIF|PDF), scope (TRANSACTIONS|BUDGET|NETWORTH|ALL), from\_date, to\_date, status, file\_ref

### **AuditLog**

- audit\_id, household\_id, actor\_user\_id, action, entity\_type, entity\_id, before\_json, after\_json, created\_at

## **4) Requisitos por módulo (Épicas → Historias → Criterios de aceptación)**

### **M01) Identidad, hogares y permisos**

#### **Épica M01-E1: Autenticación y sesión**

- US M01-01: Como usuario, quiero registrarme e iniciar sesión para acceder a mis datos.
- AC: soporte email/OTP o email/password; reset; bloqueo por intentos; sesión expira; dispositivos registrados.
- US M01-02: Como usuario, quiero ver y cerrar sesiones activas.
- AC: lista de dispositivos + “cerrar sesión” individual o global.

#### **Épica M01-E2: Hogares (household)**

- US M01-03: Como owner, creo un hogar con moneda base y config inicial.
- AC: 1 hogar por defecto; se puede crear más (según plan); moneda base no se cambia sin migración.
- US M01-04 (V2): Como owner, invito miembros con roles.
- AC: invitación expira; aceptación; auditoría; permisos por rol.

#### **Épica M01-E3: Permisos**

- US M01-05: Como owner/admin, defino roles (OWNER/ADMIN/MEMBER/VIEWER).

- AC: VIEWER solo lectura; MEMBER edita transacciones; ADMIN gestiona reglas/categorías; OWNER gestiona facturación y borrado.

## **M02) Onboarding e ingestión de datos**

### **Épica M02-E1: Setup inicial guiado**

- US M02-01: Wizard: moneda base, 3–5 cuentas iniciales, categorías base, presupuesto inicial.
- AC: completado en < 3 minutos; se puede saltar y retomar.

### **Épica M02-E2: Importación CSV/QIF (MVP)**

- US M02-02: Importar CSV con mapeo de columnas y previsualización.
- AC: mapping persistente por banco; dedupe por (fecha,monto,descripcion,account,external\_id).
- US M02-03: Importar QIF (V1.2).
- AC: soporta cuentas/transactions; mapeo a categorías; log de errores por línea.

### **Épica M02-E3: Sync bancario (V3 opcional)**

- US M02-04: Conectar una institución y sincronizar transacciones.
- AC: reintentos; manejo de desconexión; no duplica; respeta “posted\_date vs txn\_date”.

## **M03) Cuentas y monedas**

### **Épica M03-E1: CRUD de cuentas**

- US M03-01: Crear/editar/archivar cuenta.
- AC: tipos de cuenta; saldo inicial; “incluir en presupuesto” y “incluir en net worth”.
- US M03-02: Transferencias entre cuentas.
- AC: crea 2 transacciones enlazadas; conversión de moneda si aplica; balance coherente.

### **Épica M03-E2: Multi-moneda (V1.3)**

- US M03-03: Manejar transacciones en monedas múltiples y convertir a moneda base en reportes.
- AC: tasa por fecha; tasa manual; rounding control; auditoría de tasa usada.

## **M04) Transacciones (motor)**

### **Épica M04-E1: Registro y edición**

- US M04-01: Registrar gasto/ingreso/transferencia con campos clave.
- AC: validaciones (monto ≠ 0, cuenta requerida, fecha requerida).
- US M04-02: Split de transacciones.
- AC: suma de splits = monto; categoría a nivel split; reporte respeta splits.
- US M04-03: Adjuntos.
- AC: subir/visualizar; límite tamaño; export incluye referencia.

### **Épica M04-E2: Estados y conciliación**

- US M04-04 (V1.2): Estados PENDING/CLEARED/RECONCILED y conciliación por corte.
- AC: reconciliación por rango de fechas; diferencias; lock de periodo reconciliado.

#### **State machine (Transaction.status)**

- PENDING → CLEARED → RECONCILED
- PENDING/CLEARED → VOID (si se anula)

Reglas: RECONCILED bloquea edición de amount/account/date salvo “modo admin + audit”.

## **M05) Categorías, merchants, tags**

### **Épica M05-E1: Catálogo**

- US M05-01: CRUD de categorías con jerarquía (padre/hijo).
- AC: no permite ciclos; merge de categorías; reasignación masiva.
- US M05-02: Tags.
- AC: multi-tag por transacción; filtros en reportes.

### **Épica M05-E2: Merchant normalization (V1.2)**

- US M05-03: Normalizar payee\_text para agrupar por merchant.
- AC: reglas de normalización; alias; sugerencias; “set default category for merchant”.

## **M06) Reglas y revisión**

### **Épica M06-E1: Rules engine**

- US M06-01: Crear reglas “si coincide X entonces hacer Y”.
- AC: prioridad; preview (simular) antes de activar; registro de “rule applied” por transacción.
- US M06-02: Plantillas de split por merchant (V1.3).
- AC: porcentajes o montos; tolerancias; excepción manual.

### **Épica M06-E2: Needs Review (V1.2)**

- US M06-03: Cola de revisión de transacciones.
- AC: razones; asignación a miembro (si hogar); “marcar resuelto”.

## **M07) Presupuesto (3 métodos coexistentes)**

> Debe soportar al menos 1 método en MVP y extender sin romper datos.

### **M07-E1: Spending Plan (MVP)**

- US M07-01: Configurar items: ingresos, bills, subs, planned spending, savings goal.

- AC: cálculo del “restante del mes”; items recurrentes; vista por mes.
- US M07-02 (V2): Proyectar 12 meses y “Other Spend” por promedio/custom/no-projection (equivalente al comportamiento descrito en Simplifi).
- AC: selector de método; recalcular automático al cerrar mes.

#### **M07-E2: Category Budgeting (V1.1)**

- US M07-03: Presupuesto por categoría con planned\_amount.
- AC: alertas por umbral; rollover opcional; comparativo vs real.
- US M07-04 (V1.2): Rollover mensual por categoría.
- AC: carryover transparente (campo explícito); opción de reset; auditoría.

#### **M07-E3: Flex Budgeting (V1.2)**

- US M07-05: Buckets flexibles con categorías incluidas.
- AC: bucket suma categorías; no doble conteo; edición simple.

#### **M07-E4: Envelope/Zero-based (V2)**

- US M07-06: Asignar dinero disponible a categorías (available/assigned).
- AC: no asignar más que “to be assigned”; objetivos/targets; auto-assign (opcional).

### **M08) Recurrentes, bills y suscripciones (calendario)**

#### **M08-E1: Recordatorios manuales (MVP)**

- US M08-01: Crear recordatorio de bill/sub/ingreso con frecuencia.
- AC: notificaciones; lista + calendario; “marcar pagado” crea txn (opcional) o se asocia a una txn existente.

#### **M08-E2: Detección automática (V1.2)**

- US M08-02: Detectar recurrentes al importar/sincronizar transacciones.
- AC: score de confianza; requiere confirmación usuario para crear RecurringSeries si score < threshold.
- US M08-03: Instancias EXPECTED y estados (OVERDUE).
- AC: si llega transacción matching → state MATCHED/PAID; si no llega → OVERDUE; usuario puede SKIP/RESCHEDULE.

#### **M08-E3: Centro de suscripciones (V1.2)**

- US M08-04: Vista lista de suscripciones con monto, frecuencia, próxima fecha.
- AC: derivado de RecurringSeries/Instances; muestra histórico; alerta de aumento de costo.

#### **State machine (RecurringInstance.state)**

- EXPECTED → MATCHED → PAID
- EXPECTED → OVERDUE

- EXPECTED/OVERDUE → SKIPPED | RESCHEDULED | CANCELED

Reglas: una instancia no puede volver a EXPECTED si ya fue resuelta (salvo “reopen” con audit).

## **M09) Cash Flow / Forecast (proyección)**

### **M09-E1: Forecast básico (V1.1)**

- US M09-01: Proyectar saldo agregado por mes usando recurring + planned spend.
- AC: usa Schedule + SpendingPlanItems; muestra “balance estimado” por fin de mes.

### **M09-E2: Projected Cash Flow por cuenta (V2)**

- US M09-02: Gráfico diario/semanal de balance futuro hasta 1 año por cuenta.
- AC: integra ingresos/bills/subs/planned; recalcula con transacciones reales; permite filtrar cuentas incluidas.

## **M10) Metas y planificación**

### **M10-E1: Goals (V1.2)**

- US M10-01: Crear meta (target\_amount, target\_date opcional).
- AC: sugiere contribución mensual; permite contribuciones manuales; liga transacciones.
- US M10-02: Integración con Spending Plan.
- AC: item “SAVINGS\_GOAL” descuenta del restante; progreso visible.

## **M11) Deuda**

### **M11-E1: Préstamos y tarjetas (V2)**

- US M11-01: Definir préstamo con tasa, plazo, cuota y generar amortización.
- AC: tabla; recálculo si cambia extra\_payment; export.
- US M11-02: Estrategias (bola de nieve/avalancha).
- AC: ordena cuentas; muestra fecha estimada de payoff; impacto de extra\_payment.

## **M12) Patrimonio e inversiones**

### **M12-E1: Net Worth (MVP)**

- US M12-01: Calcular net worth como activos - pasivos y mostrar tendencia.
- AC: snapshots diarios/mensuales; excluye cuentas marcadas.
- US M12-02 (V1.2): Breakdown por tipo de activo/pasivo.
- AC: gráfico + tabla.

### **M12-E2: Inversiones (V2)**

- US M12-03: Holdings manuales con precio y rendimiento.

- AC: rendimiento = (precio - avg\_cost) \* qty; soporta multi-moneda; histórico de precios.

## **M13) Reportes e insights**

### **M13-E1: Reportes base (MVP)**

- US M13-01: Gastos por categoría (mes actual + comparativo vs mes anterior).
- AC: filtros por cuenta, tag, merchant; export a CSV/PDF.
- US M13-02: Cash flow mensual (income vs expense) y saving rate.
- AC: calcula saving\_rate = (income-expense)/income.

### **M13-E2: Subscriptions report (V1.2)**

- US M13-03: Lista de suscripciones y costo total mensual/anual.
- AC: variación de precio; próximos cobros.

### **M13-E3: Insights (V2)**

- US M13-04: Detección de anomalías (spike) y avisos.
- AC: define baseline por categoría/merchant; notifica solo si supera umbral y no es recurrente.

## **M14) Exportación, backup y auditoría**

### **M14-E1: Export (MVP)**

- US M14-01: Exportar transacciones a CSV (por rango y filtros).
- AC: incluye splits y tags; formato consistente; job async con estado.
- US M14-02 (V1.2): Export QIF y/o PDF (según prioridad).
- AC: incluye cuentas y categorías; PDF de reportes.

### **M14-E2: Auditoría (V1.2)**

- US M14-03: AuditLog para acciones críticas.
- AC: track de cambios; visible para owner/admin; exportable.

## **M15) Notificaciones**

### **M15-E1: Sistema de alertas**

- US M15-01: Notificaciones por: bills próximos, suscripciones, overbudget, instancias overdue.
- AC: canales (push, email); frecuencia configurable; “quiet hours”.

## **M16) Seguridad y cumplimiento (no-funcional)**

### **M16-E1: Seguridad**

- Requerido: TLS, cifrado de secretos, protección contra brute-force, control de sesión.

- Móvil: soporte PIN/biometría; bloqueo al background.
- Privacidad: política clara; minimización de datos; opción export + delete account.

#### **M16-E2: Resiliencia**

- Backups automáticos; recuperación; integridad referencial; dedupe en import-sync.

### **M17) Administración y operación**

#### **M17-E1: Panel de operación (interno)**

- Métricas: jobs de import/export, errores por conector, latencia, usuarios activos.
- Herramientas: re-procesar import, ver logs, bloquear reglas defectuosas, feature flags por versión.

## **5) Especificaciones “de borde” (para evitar fallas en producción)**

### **5.1 Dedupe de transacciones (import"sync)**

- Regla base: (account\_id, posted\_date±1, amount, normalized\_payee, external\_id si existe).
- Si match con txn existente → no crea duplicado; registra “import\_link”.
- Si conflicto → a ReviewQueueItem con reason RULE\_CONFLICT o DUPLICATE\_SUSPECT.

### **5.2 Matching recurring ↔ transacciones reales**

- Matching por ventana ( $\pm N$  días), monto (tolerancia % o fijo), merchant/payee, cuenta.
- Si match: instancia EXPECTED → MATCHED y se vincula a txn.
- Si llega la fecha y no hay match: EXPECTED → OVERDUE y dispara notificación.

### **5.3 Cálculo consistente con splits**

- Toda analítica por categoría debe usar TransactionSplit si existe; si no, Transaction.category\_id.
- Transferencias se excluyen de “gasto” y “income” por defecto.

### **5.4 Lock de periodos**

- Al cerrar mes o reconciliar, el sistema debe poder “LOCK” BudgetPeriod y/o transacciones reconciliadas.
- Cambios posteriores requieren rol ADMIN/OWNER y quedan auditados.

### **5.5 Multi-moneda**

- Toda métrica agregada debe definir: moneda base del hogar + tasa usada por fecha.
- Guardar tasa aplicada en snapshots/export para auditoría.

## **6) API (propuesta) – endpoints mínimos por versión**

> Si decides microservicios o monolito es indiferente; esto es para contratos.

## MVP

- POST /auth/register | /auth/login | /auth/logout
- POST /households (auto al registrar)
- CRUD /accounts
- CRUD /categories | /tags
- CRUD /transactions (incl. split)
- POST /imports/csv (job) + GET /imports/{id}
- GET /reports/spend\_by\_category?from&to&filters
- GET /reports/cashflow?from&to
- CRUD /recurring (manual reminders) + GET /calendar?month=YYYY-MM
- CRUD /budgets/spending-plan (items) + GET /budgets/spending-plan?month

## V1.x

- CRUD /rules + POST /rules/simulate
- GET /review-queue + PATCH /review-queue/{id}
- GET /subscriptions (derived) + PATCH /subscriptions/{id} (pause/cancel)
- GET /networth + GET /networth/snapshots
- POST /exports (job) + GET /exports/{id}

## V2

- CRUD /households/{id}/members + roles/permissions
- GET /cashflow/projected?account=...&horizon=365
- GET /debt/plans + POST /debt/plans/simulate
- CRUD /investments/holdings + /prices/import

## 7) Criterios de calidad (producción)

- Rendimiento: lista de transacciones (10k registros) paginada y con búsqueda < 300ms (server).
- Confiabilidad: import idempotente (repetir el mismo archivo no duplica).
- Seguridad: rate-limit en auth; cifrado en reposo para PII; logs sin datos sensibles.
- Observabilidad: trazas por job, métricas por endpoint, alertas por error-rate.

## 8) Anexo A – enlaces de referencia (fuentes oficiales)

> Se listan para trazabilidad del benchmark. No son requisitos; son evidencia de “qué existe” en el mercado.

- Bluecoins (Google Play): <https://play.google.com/store/apps/details?id=com.rammigsoftware.bluecoins>
- Bluecoins (sitio oficial): <https://www.bluecoinsapp.com/>

- Quicken Simplifi (Google Play): <https://play.google.com/store/apps/details?id=com.quicken.acme>
- Quicken Simplifi (producto): <https://www.quicken.com/products/simplifi/>
- Quicken Simplifi Help – Spending Plan:  
<https://support.simplifi.quicken.com/en/articles/4212702-understanding-your-spending-plan>
- Quicken – Projected Cash Flow: <https://www.quicken.com/features/projected-cashflow/>
- Monarch (pricing): <https://www.monarch.com/>
- Monarch Help – Recurring:  
<https://help.monarch.com/hc/en-us/articles/4890751141908-Tracking-Recurring-Expenses-and-Bills>
- Rocket Money pricing:  
<https://www.rocketmoney.com/learn/personal-finance/how-much-does-rocket-money-cost>
- Rocket Money manage subscriptions: <https://www.rocketmoney.com/feature/manage-subscriptions>
- Empower (Google Play): <https://play.google.com/store/apps/details?id=com.personalcapital.pcapandroid>
- NerdWallet Subscriptions & Bills FAQs: <https://support.nerdwallet.com/hc/en-us/articles/33849929735949-NerdWallet-Subscriptions-and-Bills-FAQs>
- YNAB pricing: <https://www.ynab.com/pricing>