



**PRD — Área Médica**

**Manager e Integración con Banesco**

Versión	1.0
Estado	Draft
Fecha	30/09/2025
Autor	José Cabrera (WIP)

<b>1) Resumen ejecutivo</b>	<b>2</b>
<b>2) Objetivos y No-Objetivos</b>	<b>3</b>
<b>3) Alcance funcional (MVP)</b>	<b>3</b>
<b>4) Requisitos de UX/UI</b>	<b>4</b>
<b>5) Requisitos técnicos</b>	<b>5</b>
<b>6) API (borrador de contrato)</b>	<b>6</b>
<b>7) Modelo de estados (state machine)</b>	<b>7</b>
<b>8) Datos y esquema (propuesta inicial)</b>	<b>8</b>
<b>9) Seguridad, privacidad y cumplimiento</b>	<b>8</b>
<b>10) Observabilidad y SLO</b>	<b>8</b>
<b>11) Criterios de aceptación (extracto)</b>	<b>8</b>
<b>12) Roadmap y hitos</b>	<b>9</b>
<b>13) Riesgos y mitigaciones</b>	<b>9</b>
<b>14) Supuestos (explícitos)</b>	<b>9</b>
<b>15) Anexo · Diagrama de secuencia (Mermaid)</b>	<b>9</b>
<b>16) Glosario</b>	<b>11</b>
<b>17) Checklist de “hecho” (MVP)</b>	<b>11</b>
Decisiones abiertas / Pendientes	12

## 1) Resumen ejecutivo

Desarrollar una aplicación web que gestione y consulte transferencias bancarias de clientes, integrada con Banesco. El sistema ofrece: (1) Formulario para registrar una transferencia (Transaction/Transfer ID + datos del cliente), (2) Detalle/estado de una transferencia puntual (por Transfer ID) con descarga de comprobante PDF generado desde un template y (3) Listado de todas las transferencias.

Tecnologías comprometidas:

- Diseño: Figma (UI/UX Desktop & Mobile; sin tablet).
- Frontend: React + TypeScript, Material UI.
- Backend: FastAPI (Python), PostgreSQL.
- Integración externa: API Banesco.
- Infraestructura: Dominio GoDaddy (validación + subdominio dedicado). Hosting a definir (candidato: DigitalOcean).
- Deadline MVP: octubre de 2025 (fin de mes).

## 2) Objetivos y No-Objetivos

### 2.1 Objetivos

- Registrar transferencias con validaciones de negocio.
- Consultar estado sincronizado con Banesco bajo rate limit (recomendado: 2 req/min por mismo Transfer ID).
- Generar un PDF de comprobante a partir de un template.
- Ofrecer UI Desktop/Mobile (sin tablet).
- Integrar autenticación y autorización con usuarios con permisos.

### 2.2 No-Objetivos (MVP)

- Reconciliación contable multi-banco avanzada.
- Reportería BI y dashboards analíticos.
- Soporte offline del frontend.
- Automatizar compra de dominio o DNS (solo validación y configuración de subdominio).

### 3) Alcance funcional (MVP)

#### 3.1 Vistas de producto

##### 1. Formulario de transferencia

- Campos mínimos: Nombre (alfabético), Teléfono (11 dígitos, sin "+"), Identificación (prefijo "V" + 4-8 dígitos), Concepto (texto), Referencia (numérica; validar especificación exacta de Banesco), Banco (selector: Banesco / Transferencia móvil).
- Validaciones: formato y obligatoriedad según reglas.
- Al guardar: crear/actualizar transferencia y redirigir al detalle.

##### 2. Detalle de transferencia (Transaction ID Page)

- Mostrar estado y metadatos.
- Si In progress: permitir refrescar estado (respetando rate limit).
- Si Completed/Canceled: permitir descargar PDF (template + datos de la transferencia).

##### 3. Listado de transferencias

- Tabla con filtros (por estado, fecha, referencia, teléfono).
- Paginación, ordenamiento y acceso al detalle.

#### 3.2 Flujos clave (alto nivel)

- Crear/consultar: El usuario autorizado carga el Transfer ID + datos del cliente; el backend busca en DB; si no existe, consulta Banesco, persiste y retorna; si existe, maneja estados según política; redirige al detalle.
- Actualizar estado: Cuando esté In progress, permitir polling manual con rate limit por Transfer ID.
- Generar PDF: Para Completed/Canceled, generar comprobante vía servicio de render (plantilla HTML → PDF) y servir descarga.

#### 4) Requisitos de UX/UI

- Figma como fuente de verdad de UI (design system y prototipos).
- Material UI para componentes (consistencia visual, accesibilidad, responsividad).
- Soporte de temas (claro/oscuro) e i18n listo para el futuro (ES por defecto).
- Desktop + Mobile (no Tablet). Wireframes y prototipos navegables en Figma.

Nota: Referencia de una tranferencia real



Banco: BANCO DE VENEZUELA

Número celular:  2

Identificación:  3

Concepto:  4

Fecha: 02/09/2025

Referencia:  5

## 5) Requisitos técnicos

### 5.1 Frontend (React + TS, MUI)

- Estructura por features, RTK Query o React Query para data-fetching y caché.
- Manejo de formularios con react-hook-form + zod/yup (validaciones).
- Gestión de auth con tokens (JWT) almacenados en memory store.
- Rutas protegidas (React Router).
- Paginación server-side, debounce en filtros.
- Telemetría de UI (eventos clave).

### 5.2 Backend (FastAPI + PostgreSQL)

- Arquitectura por capas (domain/application/infrastructure), Pydantic para DTOs.
- Idempotencia por Transfer ID; caching breve de respuestas Banesco por Transfer ID.
- Rate limit por Transfer ID: 2/minuto (configurable). Registrar en DB para auditoría.
- PDF service: motor de templates (Jinja2) + wkhtmltopdf/WeasyPrint/puppeteer headless (a definir en spike).
- Migrations con Alembic, CI/CD con tests (unit/integration) y linters.
- Observabilidad: logs estructurados, métricas (Prometheus), trazas (OTel).
- Seguridad: RBAC, hashing de contraseñas (Argon2/BCrypt), rotación de secretos, TLS.

Referencias:

[Diagrama de Flujo](#)

[Diagrama de secuencia](#)

[API de Banesco](#)

### 5.3 Integración Banesco (gateway)

- Cliente resiliente con time-outs, retry con backoff (solo en errores transitorios), circuit breaker.
- Mapeo de errores a códigos de dominio (p.ej., BANESCO\_TIMEOUT, BANESCO\_NOT\_FOUND, RATE\_LIMITED).
- Auditoría: registrar request/response headers mínimos + hash del payload (no PII sensible).

### 5.4 Infraestructura y dominio

- Dominio en GoDaddy: validar propiedad y configurar subdominio (ej. manager.areamedica.tld).
- Hosting: DigitalOcean como candidato; contenedores (Docker), reverse proxy (Nginx/Caddy), TLS (Let's Encrypt).
- Backups de PostgreSQL (p.ej., 7/30 días), restauración probada.

## 6) API (borrador de contrato)

Base URL: /api/v1

1) POST /transactions

Request (ejemplo):

```
{
  "transaction_id": "string",
  "customer": {
    "full_name": "string",
    "phone": "#####",
    "national_id": "V#####",
    "concept": "string"
  },
  "reference": "#####",
  "bank": "BANESCO"
}
```

Semántica: valida schema y negocio; si no existe, consulta Banesco, persiste y retorna; si existe, aplica política de estado.

Response (ejemplo):

```
{ "transaction_id": "string", "status": "IN_PROGRESS|COMPLETED|CANCELED|UNKNOWN",
  "details": {...} }
```

2) GET /transactions/{transaction\_id}

- Devuelve datos + estado actual (desde DB).

3) GET /transactions/{transaction\_id}/status

- Si IN\_PROGRESS, intenta refresh contra Banesco respetando rate limit; si COMPLETED|CANCELED, retorna desde DB.

4) GET /transactions/{transaction\_id}/pdf

- Genera/recupera PDF (si COMPLETED|CANCELED), Content-Type: application/pdf.

5) GET /transactions (listado, paginado)

- Parámetros: page, size, status, reference, phone, from, to.



Errores estándar: 400 (validación), 401/403 (authz), 404 (no existe), 409 (inconsistencia/estado), 429 (rate limit), 5xx (transitorio).

## 7) Modelo de estados (state machine)

IN\_PROGRESS → (refresh exitoso) → COMPLETED | CANCELED

IN\_PROGRESS → (timeout/transitorio) → IN\_PROGRESS

UNKNOWN → (regla de negocio) → TO\_REVIEW/RETRY (bandera y alerta interna)

Registrar eventos de estado en transaction\_events (timestamp, actor/sistema, old\_state, new\_state, razón).

## 8) Datos y esquema (propuesta inicial)

Tablas:

- users (id, email, hash, activo)
- roles (id, name) y user\_roles
- transactions: id (uuid), transaction\_id (unique), status, bank, reference, customer\_full\_name, phone, national\_id, concept, banesco\_payload (jsonb), created\_at, updated\_at
- transaction\_events (id, transaction\_id\_fk, old\_state, new\_state, reason, at)
- pdf\_artifacts (id, transaction\_id\_fk, storage\_path, checksum, created\_at)
- rate\_limit (transaction\_id, window\_start, count)

Índices: transaction\_id, status, (created\_at, status), reference, phone.

## 9) Seguridad, privacidad y cumplimiento

- PII mínima: cifrar en tránsito (TLS) y evaluar cifrado en reposo (column-level).
- RBAC granular (listar/consultar/descargar PDF).
- Auditoría de accesos (quién, cuándo, qué).
- Gestión de secretos vía variables de entorno y/o vault.
- Cumplimiento: mejores prácticas generales para protección de datos personales aplicables al contexto local.

## 10) Observabilidad y SLO

- Logs estructurados (correlation id), métricas (latencia endpoints, tasa de errores, llamadas a Banesco por estado), tracing distribuido.

- SLO (MVP): p99 latencia API < 800 ms (sin llamadas a Banesco), disponibilidad 99.5% mensual, error budget 0.5%.
- Alertas: timeouts Banesco sostenidos, explosión de 429, ratio de UNKNOWN > umbral.

## 11) Criterios de aceptación (extracto)

- Formulario: validaciones exactas de formato (tel: 11 dígitos; ID: 'V' + 4–8 dígitos; referencia numérica). Guardar → redirigir a detalle.
- Detalle: botón 'Actualizar' visible solo si IN\_PROGRESS y bloqueado cuando el rate limit se alcanzó (feedback al usuario).
- PDF: descargable solo si COMPLETED/CANCELED; contiene campos clave de la transferencia formateados.
- Listado: filtros por estado/fecha/teléfono/referencia con paginación server-side.
- Dominio: subdominio activo y con TLS válido; redirección HTTP→HTTPS.

## 12) Roadmap y hitos

- Semanas 1–2: Diseño en Figma (flows + UI), setup repo, scaffolding FE/BE, auth básica.
- Semana 3: Integración Banesco (gateway + mapping + tests), esquema DB, migraciones.
- Semana 4: PDF template + render service, listado + filtro, observabilidad base.
- Semana 5: Endurecimiento (rate limit, retries, errores), hardening seguridad, dominio/subdominio y TLS.
- Fin de octubre 2025: MVP listo para UAT.

## 13) Riesgos y mitigaciones

- Especificación Banesco incompleta o cambiante → Spike técnico temprano + contrato tolerante (feature flags).
- Rate limit insuficiente → Ajuste dinámico y cache por Transfer ID.
- Convivencia con usuarios legacy → Plan de sincronización o federación de identidades (migración progresiva).
- PDF rendering frágil → Tests visuales del template y fallback a pre-generado.

## 14) Supuestos (explícitos)

- El formato de Referencia de Banesco es numérico y de 9 dígitos (a validar).
- La limitación de 2/min por Transfer ID es aceptada por negocio en MVP.
- Solo Desktop/Mobile en la primera release.

## 16) Glosario

- Transfer/Transaction ID: Identificador único de transferencia.
- Rate limit: Restricción de frecuencia de llamadas (p.ej. 2/min por Transfer ID).
- RBAC: Control de acceso basado en roles.
- UAT: User Acceptance Testing.

## 17) Checklist de “hecho” (MVP)

- ☐ Figma listo (flows + UI Desktop/Mobile).
- ☐ Vistas: Formulario / Detalle / Listado.
- ☐ Validaciones de campos implementadas.
- ☐ Integración Banesco con manejo de errores y rate limit.
- ☐ Generación de PDF con template y datos reales.
- ☐ Autenticación + permisos (RBAC).
- ☐ Observabilidad base (logs, métricas, trazas).
- ☐ Dominio/subdominio operativo con TLS.
- ☐ Plan de pruebas + casos de aceptación aprobados.
- ☐ Deploy MVP para UAT antes de fin de octubre 2025.

## Decisiones abiertas / Pendientes

- Formato exacto de referencia Banesco y documentación oficial.
- Selección de motor PDF (wkhtmltopdf vs WeasyPrint vs headless browser).
- Hosting final (DO u otro) y sizing inicial.
- Estrategia de identidad (convivencia con usuarios legacy vs SSO).