Sistema de Banca por Internet (BP)

1. Resumen ejecutivo y alcance

El sistema de Banca por Internet (BP) en Ecuador se concibe como una plataforma digital integral que permita a los clientes acceder de manera segura, rápida y sencilla a los servicios financieros más relevantes: consulta de saldos y movimientos, transferencias entre cuentas propias e interbancarias, pagos de servicios, notificaciones transaccionales y un proceso de onboarding digital con biometría facial.

Objetivo del sistema

- Facilitar la inclusión financiera al ofrecer un canal digital accesible 24/7 desde web y móvil.
- Mejorar la experiencia del cliente mediante tiempos de respuesta bajos, disponibilidad alta y procesos intuitivos.
- Reducir costos operativos asociados a agencias físicas, migrando transacciones al canal digital.
- Cumplir con la normativa ecuatoriana (Superintendencia de Bancos, LOPDP, BCE) y estándares internacionales de seguridad.

Stakeholders clave

- Negocio: responsables de la estrategia digital y la propuesta de valor.
- Riesgos/Fraude: aseguramiento contra suplantación de identidad, fraude transaccional y cumplimiento normativo.
- Canales: equipos de experiencia digital para web y móvil.
- Seguridad: encargados de ciberseguridad, monitoreo y respuesta a incidentes.
- Datos: administradores de bases de datos, gobierno de datos y analítica.
- Operaciones: soporte técnico, continuidad del negocio y monitoreo de disponibilidad.

Alcance vs. fuera de alcance

- **Alcance inicial (MVP):**
  - SPA Web y Aplicación móvil (framework multiplataforma).
  - Autenticación y autorización con OAuth2.0/OIDC.
  - Consultas de saldos y movimientos.
  - Transferencias propias e interbancarias.
  - Pagos básicos (servicios y préstamos).
  - Notificaciones en al menos 2 canales (email + SMS/Push).
  - Onboarding con biometría facial y validación documental.
  - Base de auditoría de operaciones.

- **Futuras fases (fuera del MVP, pero planificadas):**
  - Pagos avanzados (tarjetas de crédito, impuestos, seguros).
  - Integración con open finance y APIs externas.
  - Servicios de inversión y gestión patrimonial.
  - Analítica avanzada en tiempo real para detección de fraude.

Justificación de decisiones iniciales

1. Uso de OAuth2.0/OIDC para autenticación:
   a. Alternativa evaluada: autenticación propietaria (más compleja de mantener, menos segura).
   b. Decisión: estándar abierto para interoperabilidad y cumplimiento con NIST 800-63.
2. Arquitectura en AWS:
   a. Alternativa: infraestructura on-premise (menos escalable, mayor costo inicial, más lenta en time-to-market).
   b. Decisión: nube pública (AWS) por resiliencia, escalabilidad, servicios financieros certificados y cumplimiento (PCI DSS, ISO 27001).
3. SPA + App móvil multiplataforma:

a. Alternativa: desarrollo móvil nativo (más costoso, mayor tiempo de mantenimiento).
b. Decisión: SPA + framework multiplataforma (Flutter/React Native) para reducir costos y acelerar el time-to-market.

En resumen, el alcance inicial busca probar la solución (POV/MVP) en un entorno regulado y seguro, habilitando una rápida adopción digital y preparando la plataforma para crecer hacia un ecosistema completo de banca digital.

2. Requisitos Funcionales

El diseño funcional del sistema parte de los casos de uso más críticos en la banca digital moderna, alineados con las expectativas de clientes y con los requerimientos regulatorios en Ecuador.

Casos de uso principales

1. **Autenticación y autorización (OAuth2/OIDC):**
   a. Flujo recomendado: Authorization Code + PKCE para SPA y móvil (AWS Cognito)
   b. Integración con MFA adaptativo (OTP/FIDO2) y step-up según importe de transacción.
   c. Alternativas evaluadas: autenticación propietaria (más difícil de auditar, menos interoperable).
   d. Justificación: cumplimiento con NIST 800-63 y compatibilidad con regulaciones de la SB.

2. **Consulta de saldos y movimientos:**
   a. Acceso rápido (<2 segundos promedio) a información consolidada de cuentas.
   b. Implementación de caché en Redis para reducir latencia en consultas frecuentes.
   c. Justificación: reduce carga sobre Core Bancario y mejora experiencia del cliente.

3. **Transferencias (propias e interbancarias):**
   a. Propias: entre cuentas del mismo cliente, confirmación en tiempo real.
   b. Interbancarias: integración con servicios del Banco Central del Ecuador (BCE) para transferencias interbancarias inmediatas.
   c. Manejo de idempotencia y prevención de duplicados.
   d. Justificación: minimizar riesgo de errores operativos y cumplir con normativa de pagos del BCE.

4. **Pagos (servicios, tarjetas, préstamos):**
   a. Registro de convenios de pago y proveedores.
   b. Ejecución inmediata o programada.
   c. Conciliación automática con Core Bancario.
   d. Justificación: modernización de servicios y reducción de costos operativos por conciliación manual.

5. **Notificaciones multi-canal:**
    a. Canales soportados: email (SES), WhatsApp (Twilio/Meta).
    b. Configuración de preferencias por cliente (opt-in/opt-out, horarios de silencio).
    c. Justificación: cumplir normativa de consentimiento informado y mejorar experiencia de usuario.

6. **Onboarding con reconocimiento facial + KYC:**
    a. Captura biométrica + prueba de vida.
    b. Validación contra documento de identidad (Registro Civil, listas de prevención de lavado de activos).
    c. Justificación: reducir fraude de identidad y cumplir con normativa AML/CFT.

7. **Administración de dispositivos y sesiones:**
    a. Gestión de dispositivos confiables, cierre de sesiones remotas.
    b. Alertas en caso de inicio de sesión desde dispositivo desconocido.
    c. Justificación: control de seguridad adicional exigido por la SB.

Integraciones

- Core Banking:
    - Fuente principal de datos de productos, saldos y movimientos.
    - Implementación mediante patrones anti-corrupción (ACL) para aislar dominios.
- Sistema complementario:
    - Datos enriquecidos (ej. listas negras, scoring de fraude, información de clientes de terceros).
    - Ejemplo: integración con buro de crédito o sistema externo de KYC.

Beneficios clave

- Cliente: acceso rápido, seguro y confiable a sus finanzas.
- Banco: reducción de fraude, cumplimiento regulatorio, mayor adopción digital.
- Regulador: cumplimiento con normativa vigente (SB, BCE, LOPDP).

3. Requisitos no funcionales (NFR)

Los NFR definen el estándar de calidad del sistema. Se establecen metas **POV/MVP** y **Producción** para facilitar la evolución.

a. Disponibilidad y SLO

- Objetivo de disponibilidad (app completa):
    - POV/MVP: ≥ **99.5%** mensual.
    - Producción: ≥ **99.9%** mensual (servicios críticos ≥ **99.95%**).
- **Error budget mensual (prod 99.9%)**: ≈ 43 min/mes.

- **Mecanismos:** Multi-AZ obligatorio, health checks, autoscaling, graceful degradation.

b. Rendimiento y latencia objetivo

- **Inicio de sesión (p95):** POV ≤ 1.5s / Prod ≤ **800 ms**.
- **Consulta de saldos (p95):** POV ≤ 1.2s / Prod ≤ **500 ms** (caché Redis + lectura optimizada).
- **Transferencia propia (p95, confirmación UI):** POV ≤ 2.5s / Prod ≤ **1.2s**.
- **Transferencia interbancaria (p95):** Prod ≤ **2.5s** (dependencias BCE fuera de control: mostrar estado asíncrono y recibo).

c. Escalabilidad y elasticidad

- Dimensionamiento elástico en cómputo (EKS/ECS/Lambda) y colas (SQS).
- **POV:** cargas pico de 200 RPS.
- **Prod:** escalar a 1,000–2,000 RPS en campañas; prueba de stress 2× pico.
- **Back-pressure** y **rate limiting** por cliente/IP/usuario.

d. Confiabilidad, consistencia e idempotencia

- **Consistencia:** fuerte en transacciones (Aurora), eventual en proyecciones (CQRS + caché).
- **Idempotencia:** claves por operación (transferencias/pagos) con TTL en DynamoDB.

e. Observabilidad

- **Métricas (golden signals):** latencia, tráfico, errores, saturación.
- **Trazas distribuidas:** OpenTelemetry + X-Ray
- **Logs estructurados** (JSON) con correlación
- **SLO dashboards** (CloudWatch/Grafana)

f. Seguridad (baseline NFR)

- **Cifrado en tránsito** TLS 1.2+; **en reposo** con KMS (S3, Aurora, DynamoDB, logs).
- **Segregación de redes:** subredes privadas, endpoints VPC, SG restrictivos.
- **Gestión de secretos:** Secrets Manager/SSM; rotación.
- **WAF/Shield** y Bot Control en el perímetro.

g. DR y continuidad (RTO/RPO)

- **POV:** backups diarios + PITR (Aurora/DynamoDB), **RTO ≤ 4 h**, **RPO ≤ 1 h**.
- **Producción:** Multi-AZ + réplicas;
  - **Active/Passive multi-región (opcional): RTO ≤ 30 min**, **RPO ≤ 5 min**.
  - DNS failover con Route 53 health checks.

h. Operabilidad y soporte

- **Despliegues** blue/green o canary con rollback automático.

- **MTPR** (tiempo medio para recuperar) objetivo: ≤ 15 min (incidentes app); **MTTD** ≤ 5 min (detección vía alertas).

4. Supuestos, restricciones y riesgos

El éxito de la arquitectura depende de varios **supuestos de base**, el cumplimiento de **restricciones externas** y la gestión activa de **riesgos identificados**.

a. Supuestos técnicos y de negocio
   i. **Conectividad estable** entre los sistemas del banco (Core, sistemas complementarios) y AWS vía VPN o Direct Connect.
   ii. **Adopción del canal digital** por al menos el 40% de los clientes activos en los primeros 12 meses.
   iii. **Capacidad de integración** del Core Bancario con APIs expuestas (REST/ SOAP) y posibilidad de evolucionar hacia un ESB o event bus.
   iv. **Disponibilidad de personal especializado** en AWS, DevSecOps y ciberseguridad bancaria.
   v. **Regulador (SB, BCE)** acepta despliegue en nube pública siempre que se cumplan controles de seguridad y localización de datos cuando aplique.

b. Restricciones
   i. Legales:
   - Cumplimiento obligatorio de la **Ley Orgánica de Protección de Datos Personales (LOPDP)** en Ecuador.
   - Normativa de la **Superintendencia de Bancos (SB)** sobre seguridad de la información y continuidad.
   - Normativa del **Banco Central del Ecuador (BCE)** sobre pagos y transferencias interbancarias.

   ii. Tecnológicas:
   - Algunas dependencias del Core Bancario son sistemas legados con tiempos de respuesta variables (>2s).
   - Integraciones con BCE sujetas a disponibilidad y protocolos actuales (ej. SOAP/XML).

   iii. Presupuesto:
   - Para el POV, se limita el gasto mensual en infraestructura AWS a un minimo posible con ayuda de arquitecturas a demanda.
   - En producción, el presupuesto deberá escalar, pero con métricas FinOps y optimización continua.

   iv. Plazos: (Tiempo propuesto dispuesto a variación)
   - POV debe estar disponible en **6 meses**.
   - Producción full-feature en un plazo de **18–24 meses**.

c. Riesgos y mitigaciones

Riesgo	Severidad	Probabilidad	Mitigación
Fraude de identidad (onboarding, accesos)	Alta	Media	MFA obligatorio, biometría *liveness*, monitoreo antifraude en tiempo real.
Caída de Core Bancario	Alta	Media	Circuit breaker, caché de última consulta, colas con reintentos, DRP de Core.
Latencia alta en BCE para transferencias interbancarias	Media	Alta	Manejo asíncrono de confirmaciones, notificación posterior, SLA comunicados al cliente.
Fuga de datos sensibles (PII, biometría)	Alta	Baja	Cifrado E2E, KMS, controles de IAM, auditoría de accesos, retención mínima de biometría.
Sobrecostos en AWS	Media	Media	Uso de Savings Plans, límites presupuestales, monitoreo con AWS Budgets y alarmas.
Resistencia cultural interna (usuarios internos del banco)	Media	Media	Plan de capacitación, gestión del cambio, pilotos internos antes de liberar a clientes.
Regulador cambia requisitos (ej. localización de datos)	Alta	Baja	Diseñar multi-región con opción de almacenar en S3 en región compatible, revisión legal continua.
Ataques DDoS o bots	Alta	Alta	AWS WAF, Shield Advanced, rate limiting, scrubbing centers de ISP.

d. Conclusiones del análisis de riesgos

- El mayor impacto proviene de **seguridad y cumplimiento** (fraude, fuga de datos, ataques DDoS).
- El mayor nivel de probabilidad está en **dependencias externas** (Core y BCE).
- Mitigaciones incluyen tanto **controles técnicos** (circuit breaker, cifrado, WAF) como **organizacionales** (capacitación, gestión del cambio, comunicación con regulador).

5. Estrategia de plataforma en AWS

La plataforma de Banca por Internet se construirá sobre AWS siguiendo buenas prácticas de arquitectura multi-cuenta, seguridad desde el diseño e infraestructura como código (IaC). Esto garantiza escalabilidad, cumplimiento regulatorio en Ecuador y resiliencia ante fallas.

a. Multi-cuenta (Landing Zone)

- **Estructura de cuentas**:
  - **Prod:** cargas en producción con controles más estrictos.
  - **QA/Staging:** pruebas de calidad y pre-producción.
  - **Dev:** entornos de desarrollo aislados.
  - **Seguridad:** cuentas centralizadas para GuardDuty, Security Hub, auditoría.
  - **Datos/Analítica:** procesamiento de logs y analítica avanzada.
- **Control centralizado:** AWS Organizations con Service Control Policies (SCPs).
- **Justificación:** mejora de seguridad y gobernanza; aislamiento de riesgos entre ambientes.

b. Red y conectividad

- **VPCs dedicadas** por ambiente, con subredes públicas, privadas y de datos.
- **Subredes privadas** para microservicios y bases de datos.
- **Endpoints VPC** (S3, DynamoDB, Secrets Manager, KMS) para evitar tráfico a internet.
- **Conexión on-premise:** AWS Site-to-Site VPN en POV; migración a **Direct Connect** para producción con latencia estable y cumplimiento BCE.
- **Seguridad perimetral:** WAF + Shield Advanced en CloudFront/ALB.
- **Justificación:** cumplimiento de requerimientos de la SB sobre redes segregadas y resilientes.

c. Estrategia IaC (Infrastructure as Code)

- **Herramientas:** Terraform para infraestructura y AWS CDK para componentes específicos.
- Prácticas:
  - Repositorios separados por ambiente.
  - Validación de cambios con pipelines (CI/CD).
  - Escaneo de seguridad IaC (tfsec, cdk-nag).
- **Naming y etiquetado:** convención uniforme con campos, ayuda de Helpers DRY
- **Justificación:** reproducibilidad, auditoría y reducción de errores manuales.

d. Gestión de identidades y accesos

- **Clientes (CIAM):** Amazon Cognito User Pools con federación a Google/Microsoft/Apple (OIDC/SAML).
- **Colaboradores (Workforce):** AWS IAM Identity Center (ex SSO) federado con Microsoft Entra ID.
- **Principio de menor privilegio:** IAM Roles con políticas gestionadas; SCPs en Organization.
- **Gestión de secretos y claves:** AWS Secrets Manager para credenciales y certificados; AWS KMS/HSM para cifrado en reposo y llaves maestras.
- **Rotación automática:** contraseñas, llaves y certificados rotados periódicamente.
- **Justificación:** cumplimiento con LOPDP y regulaciones bancarias de Ecuador, evitando exposición indebida de credenciales.

e. Observabilidad y gobierno

- **Logs centralizados:** CloudWatch Logs enviados a cuenta de seguridad.
- **Monitoreo:** CloudWatch metrics, X-Ray, OpenTelemetry → Grafana.
- **Alertas:** integradas con SNS/Slack/Teams.
- **Gobierno:** tagging obligatorio, CMDB sincronizado con AWS Config.

f. Beneficios de la estrategia AWS

- **Seguridad:** cuentas aisladas + SCPs + monitoreo central.
- **Escalabilidad:** redes y servicios listos para crecer de POV a producción masiva.
- **Cumplimiento:** facilita auditorías de SB, LOPDP y estándares internacionales.
- **FinOps:** costos controlados con presupuestos, Savings Plans y visibilidad por etiquetas.

6. Diagrama de Contexto

Propósito: Mostrar el ecosistema completo que rodea a la Banca por Internet (BP) en Ecuador, identificando actores humanos y sistemas externos, límites del sistema y principales flujos de información.

6.1 Actores principales

- Cliente Web (navegador moderno).
- Cliente Móvil (app Android/iOS).
- Backoffice (operadores del banco para soporte y monitoreo).
- Administrador de Seguridad (SOC/Equipo de ciberseguridad).

6.2 Sistemas externos

- Core Bancario (saldos, movimientos, órdenes de pago/transferencia).
- Sistema Complementario/KYC (verificación de identidad, listas AML/CFT, buró de crédito).
- IdP / Proveedor de Identidad (Cognito/IdP federado con Microsoft/Google/Apple según capítulo 9).
- BCE (servicios de pagos/transferencias interbancarias).
- Proveedores de notificación (Email/SMS/Push/WhatsApp).
- Herramientas de observabilidad (SIEM/Logging centralizado, ticketing NOC/SOC).

6.3 Límite del sistema (BP)

Dentro del límite (responsabilidad directa del proyecto):

- Canales de presentación (Web y Móvil), BFFs, APIs internas, servicios de dominio (Clientes, Cuentas, Movimientos, Transferencias, Pagos, Notificaciones, Auditoría), y almacenes de datos propios (operacionales y de auditoría).

Fuera del límite (integraciones):

- Core Bancario, KYC/Complementario, BCE, IdP externo y proveedores de notificación.

6.4 Flujos de alto nivel (qué se intercambia)

1. Autenticación y autorización: Cliente ↔ IdP (OIDC/OAuth2 con PKCE). BP verifica tokens en cada solicitud.
2. Consultas de información: BP ↔ Core Bancario (saldos/movimientos). Caché de lectura para mejorar latencia.
3. Órdenes transaccionales: BP → Core (transferencias propias/pagos) y BP ↔ BCE (interbancarias). Confirmación síncrona o asíncrona según el caso.
4. Onboarding biométrico/KYC: Cliente ↔ BP ↔ Sistema Complementario (liveness, OCR, listas AML/CFT). Solo persistir evidencias mínimas requeridas por regulación.
5. Notificaciones: BP → Proveedores (email/SMS/push/WhatsApp) según preferencias del cliente, con registro de auditoría.
6. Observabilidad y seguridad: BP → SIEM/Logging/SOC (eventos, alertas, auditoría, cumplimiento).

6.5 Límites de confianza y datos personales

- PII y biometría se tratan con minimización y propósito específico (LOPDP). Se documentan bases legales (consentimiento explícito para biometría) y tiempos de retención.
- Tokens y secretos nunca viajan por canales inseguros; TLS 1.2+ extremo a extremo.
- Separación de dominios: autenticación (IdP) aislada de la lógica transaccional.

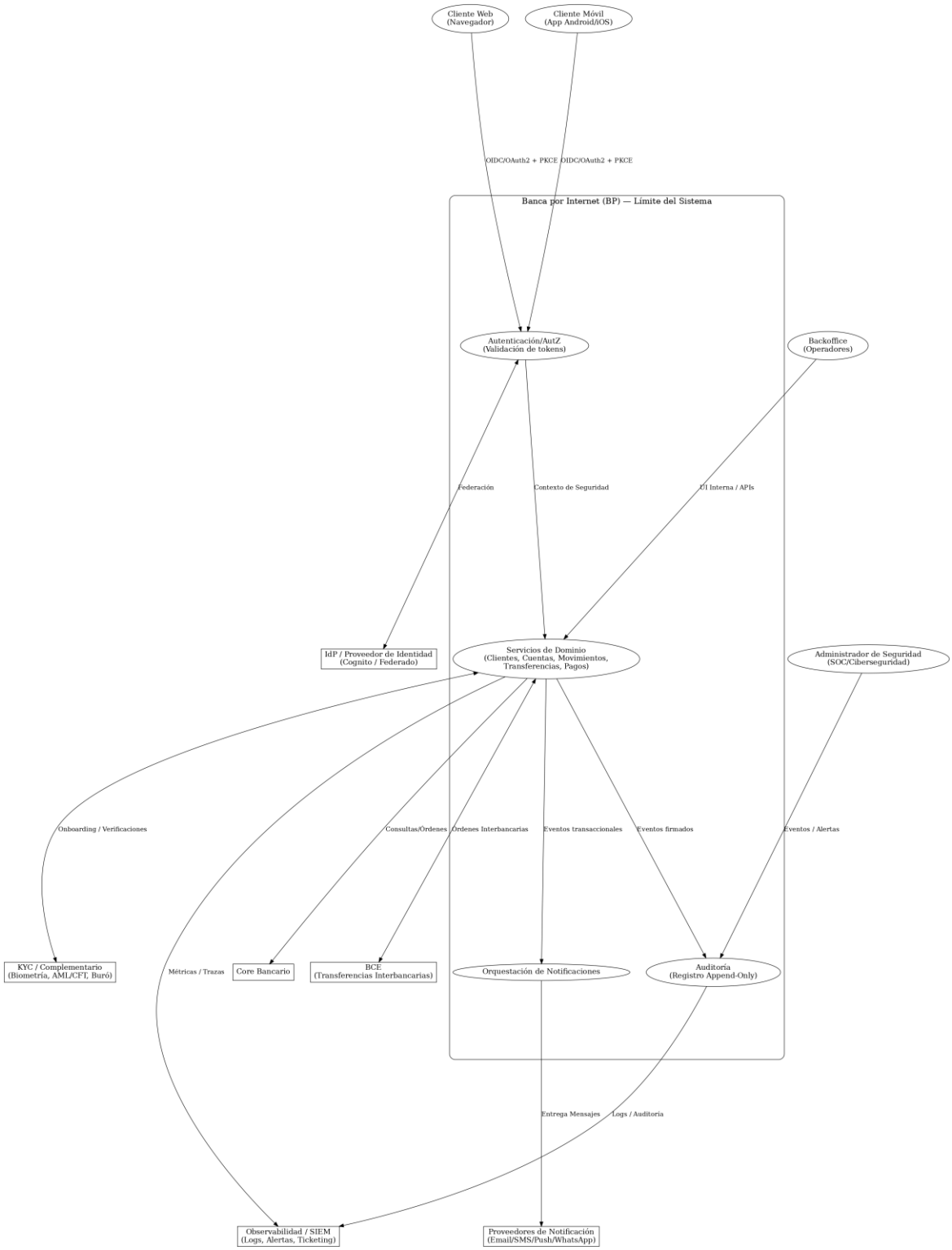
6.6 Amenazas y controles de nivel contexto

- Suplantación de identidad → MFA/step-up, liveness, device fingerprinting.
- Exposición de datos → cifrado en tránsito/represo, política de acceso mínimo, registro y auditoría.
- Indisponibilidad de dependencias (Core/BCE) → circuit breaker, colas y compensaciones.
- Abuso automatizado (bots) → WAF/Bot Control, rate limiting, detección de anomalías.

6.7 Métricas clave a nivel contexto

- Disponibilidad canal (web/móvil), tiempo de login, latencia consultas, éxito de transferencias, tasa de fraude y entregabilidad de notificaciones.

Diagrama:


Cliente Web
(Navegador)
Cliente Móvil
(App Android/iOS)
OIDC/OAuth2 + PKCE
OIDC/OAuth2 + PKCE
Banca por Internet (BP) — Límite del Sistema
Autenticación/AutZ
(Validación de tokens)
Backoffice
(Operadores)
Federación
Contexto de Seguridad
UI Interna / APIs
IdP / Proveedor de Identidad
(Cognito / Federado)
Servicios de Dominio
(Clientes, Cuentas, Movimientos,
Transferencias, Pagos)
Administrador de Seguridad
(SOC/Ciberseguridad)
Onboarding / Verificaciones
Consultas/Ordenes
Ordenes Interbancarias
Eventos transaccionales
Eventos firmados
Eventos / Alertas
KYC / Complementario
(Biometría, AML/CFT, Buró)
Métricas / Trazas
Core Bancario
BCE
(Transferencias Interbancarias)
Orquestación de Notificaciones
Auditoría
(Registro Append-Only)
Entrega Mensajes
Logs / Auditoría
Observabilidad / SIEM
(Logs, Alertas, Ticketing)
Proveedores de Notificación
(Email/SMS/Push/WhatsApp)

7. Diagrama de Contenedores

A continuación se presentan **dos vistas complementarias** de contenedores sobre AWS: una para **POV/MVP (serverless, bajo costo)** y otra para **Producción (alta disponibilidad, observabilidad completa y DR)**.

7.1 POV / MVP (Serverless)

Objetivo: minimizar costo fijo y acelerar el tiempo de salida.

Contenedores clave:

- Front: SPA (React/Amplify + S3 + CloudFront) y App móvil (Flutter).
- Perímetro: AWS WAF + Shield.
- Integración: API Gateway con Authorizer JWT (Cognito User Pool).
- Lógica: Lambda BFF Web/Móvil y Lambdas por dominio (consultas y transacciones).
- Datos: Aurora PostgreSQL Serverless v2 (RDS Proxy), DynamoDB (idempotencia/outbox), Redis (caché/rate-limit), S3 (evidencias/estáticos).
- Mensajería/eventos: SQS / EventBridge.
- Identidad: Cognito (federación OIDC/SAML con Microsoft/Google si aplica).
- Notificaciones: Pinpoint/SES (SMS/Push/Email/WhatsApp).
- Onboarding: Rekognition Face Liveness + integración KYC.
- Observabilidad: CloudWatch + X-Ray, dashboards en Grafana.

Justificación (2+ decisiones):

- Lambda + API Gateway reduce costos y complejidad en POV (pago por uso) frente a contenedores gestionados.
- Aurora Serverless v2 ofrece ACID sin sobredimensionar; alternativa evaluada: RDS provisionado (más costo fijo en POV).
- DynamoDB para idempotencia/outbox simplifica deduplicación; alternativa: tablas auxiliares en RDS (más lock contention).

7.2 Producción (Alta disponibilidad)

Objetivo: robustez, control fino de redes y observabilidad avanzada.

Contenedores clave:

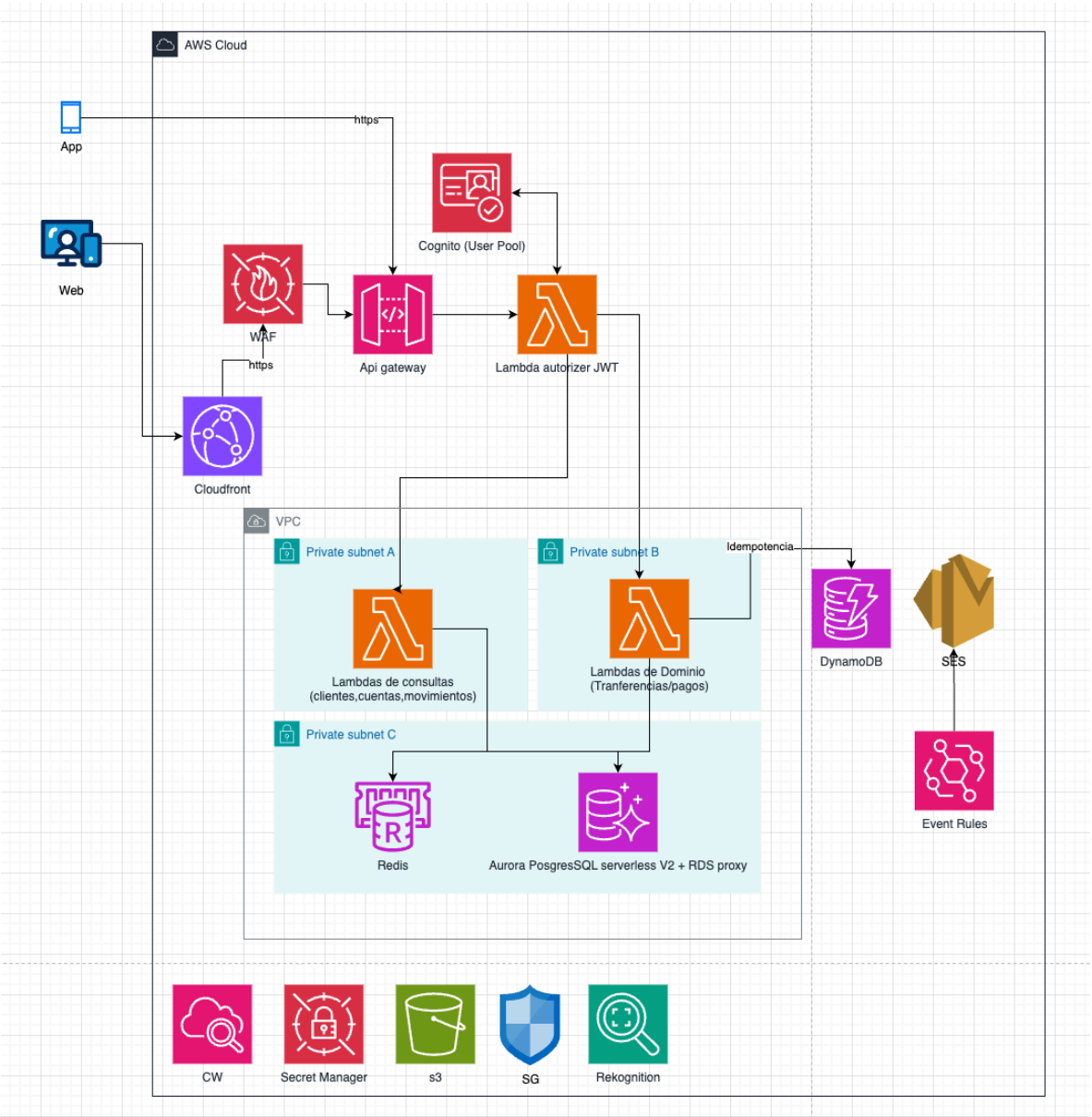
- Front: SPA (React/Amplify + S3 + CloudFront) y App móvil.
- Perímetro: WAF + Shield.
- Ingreso: ALB / API Gateway (según patrón).
- Lógica: Amazon EKS (o ECS Fargate) con BFF y microservicios (Transferencias, Cuentas/Mov., Notificaciones, Auditoría).
- Datos: Aurora PostgreSQL Multi-AZ con RDS Proxy y Read Replicas; DynamoDB (idempotencia/outbox); ElastiCache Redis (caché/rate-limit); S3 con Object-Lock (WORM).

- Asíncrono: SQS/EventBridge para integración desacoplada.
- Búsqueda/observabilidad: ElasticSearch (opcional); OpenTelemetry + X-Ray.
- Identidad/Onboarding/Notificaciones: Cognito, Rekognition, SES.
- Seguridad y secretos: KMS, Secrets Manager, políticas IAM de mínimo privilegio.

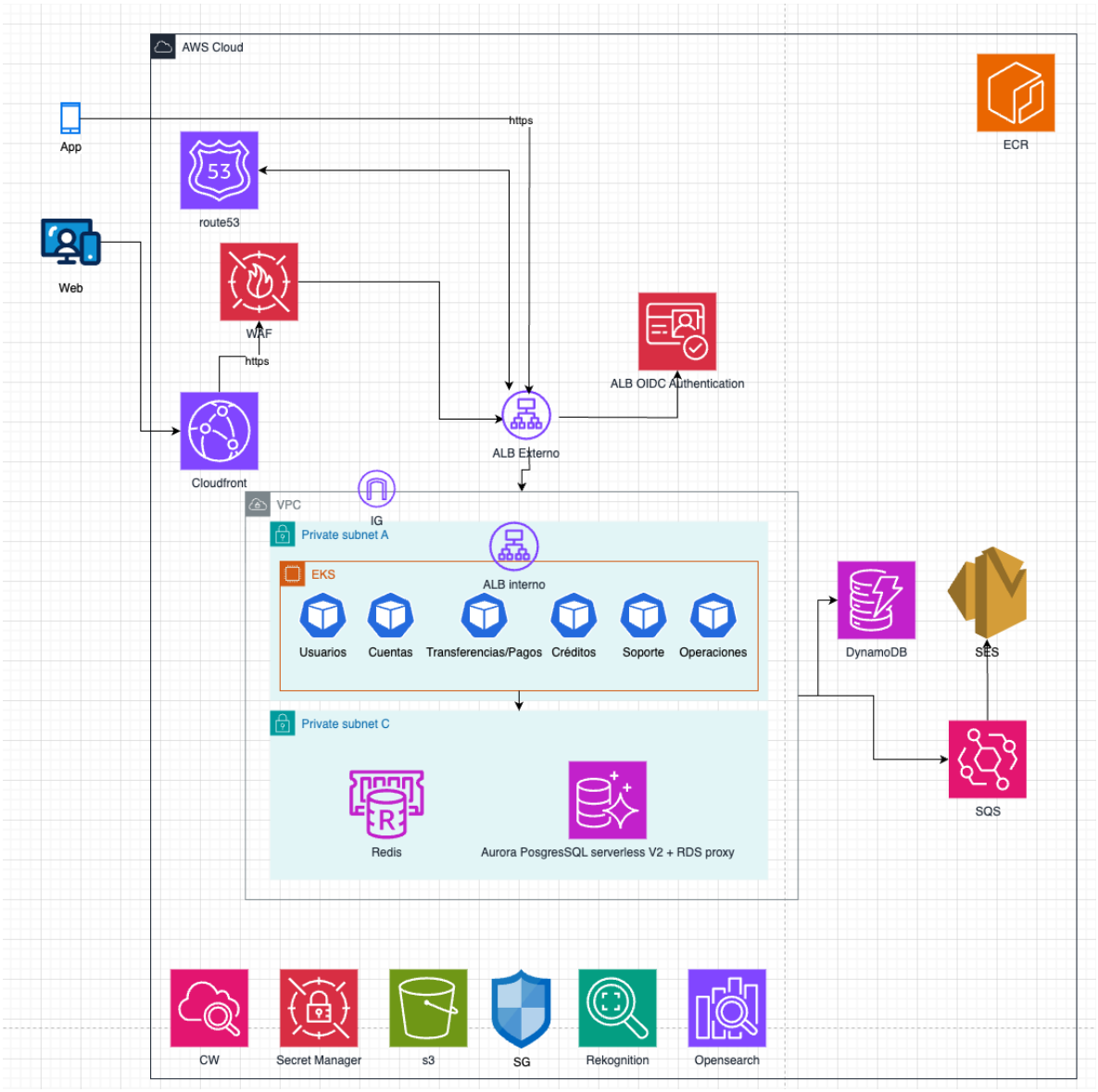
Justificación (2+ decisiones):

- EKS aporta control de red (NetworkPolicies), sidecars, afinidad de pods y malla de servicio con mTLS; alternativa: ECS Fargate (menos control, menos operación, puede ser válido si se prioriza simplicidad).
- Aurora Multi-AZ + réplicas garantiza HA y escalado de lectura; alternativa: sharding/particionamiento temprano (se difiere hasta necesitarlo).
- WAF + Shield con reglas administradas protege contra OWASP Top-10 y DDoS; alternativa: dispositivos on-prem (no aplican en nube pública).
- **7.3 Diferencias clave POV vs Producción**

Aspecto	POV/MVP	Producción
Cómputo	Lambda (BFF + dominios)	EKS/ECS para microservicios persistentes
API Ingreso	API Gateway	ALB/API GW (según servicio)
Base de datos	Aurora Serverless v2	Aurora Multi-AZ + réplicas + RDS Proxy
Idempotencia/Outbox	DynamoDB	DynamoDB
Caché	Redis opcional	Redis obligatorio para p95
Observabilidad	CloudWatch + X-Ray básico	CloudWatch + X-Ray + Grafana/Alertas avanzadas
Seguridad	WAF/Shield básicos	WAF/Shield + mTLS/mesh + políticas OPA
Costos	Pago por uso, bajo fijo	Optimizado por volumen (Savings Plans/RI)



Arquitectura PROD


AWS Cloud
App
https
ECR
route53
Web
WAF
https
ALB OIDC Authentication
Cloudfront
ALB Externo
IG
VPC
Private subnet A
ALB interno
EKS
Usuarios
Cuentas
Transferencias/Pagos
Créditos
Soporte
Operaciones
DynamoDB
SES
Private subnet C
SQS
Redis
Aurora PosgresSQL serverless V2 + RDS proxy
CW
Secret Manager
s3
SG
Rekognition
Opensearch

8. Diagramas de Componentes

En este capítulo bajamos al nivel de componentes de los dominios críticos. Se incluyen diagramas con letras grandes para impresión en vertical (A4).

8.1 BFF Web y BFF Móvil

Responsabilidades

- Composición de datos para cada canal (evitar múltiples llamadas desde el front).
- Adaptación y estabilidad de contratos hacia las apps.
- Aplicación de políticas de seguridad de borde (validación de JWT, rate-limit por usuario/IP, anti-replay).
- No contiene reglas de negocio bancario.

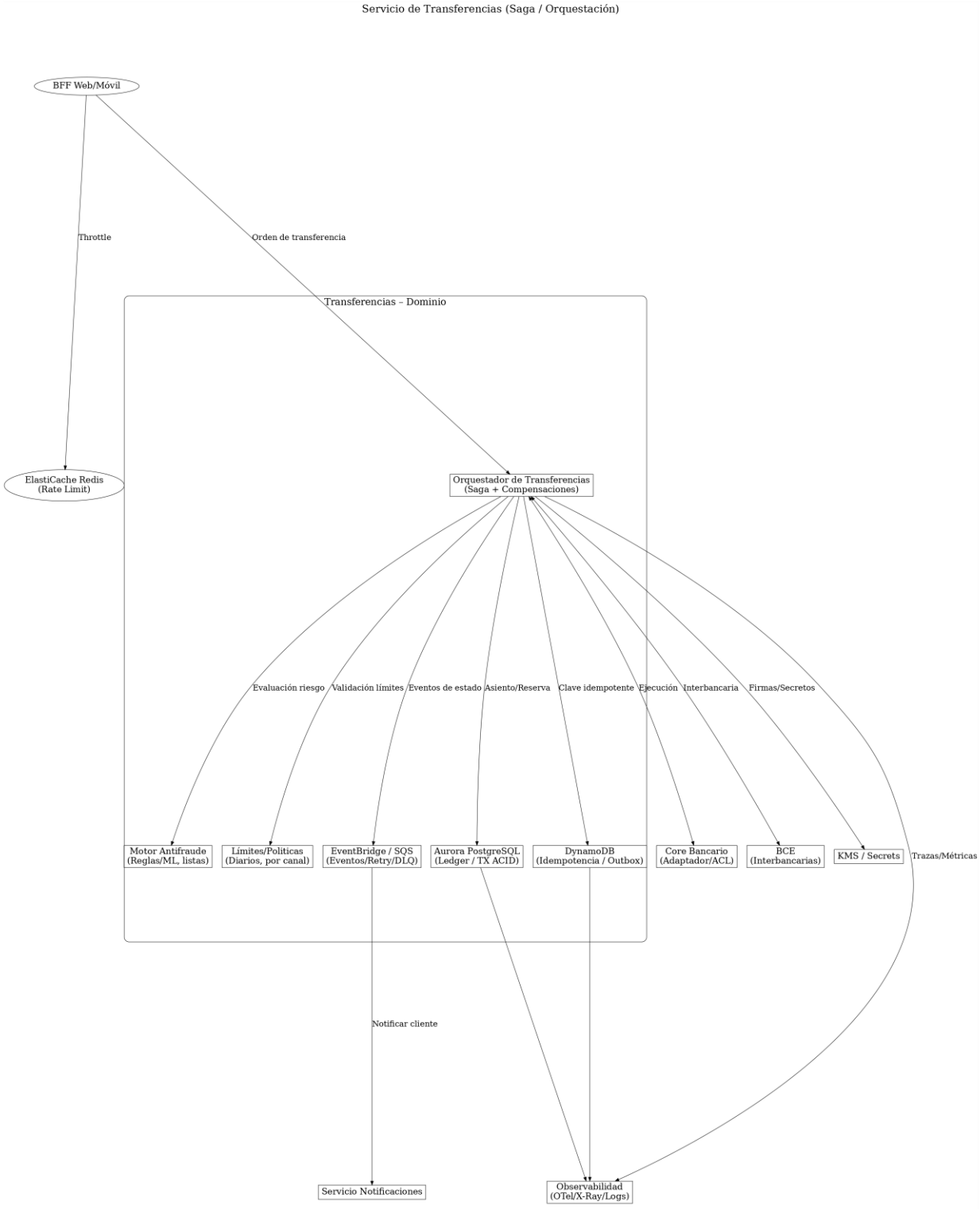
Decisiones y alternativas

- BFF por canal (Web/Móvil) vs BFF único → Se elige por canal para optimizar experiencias y aislar cambios. Alternativa: uno solo (menos despliegues, pero contratos más genéricos y pesados).
- REST vs GraphQL → Se elige REST por auditoría y control de superficies; GraphQL posible en consultas internas con persistencia de queries.

Interfaces (ejemplos)

- GET /v1/accounts/{id}/balance
- GET /v1/accounts/{id}/transactions?from=...&to=...
- POST /v1/transfers
- POST /v1/payments

8.2 Servicio de Transferencias (Saga/Orquestación)



Responsabilidades

- Orquestar transferencias propias e interbancarias.
- Enforzar límite por usuario/canal y políticas antifraude.
- Garantizar idempotencia y compensaciones en errores parciales.
- Publicar eventos de estado (iniciada, ejecutada, fallida) y generar notificaciones.

Patrones

- Saga orquestada, Transactional Outbox, Circuit Breaker, Retry con backoff.

Datos

- Aurora PostgreSQL: órdenes, asientos, reservas.
- DynamoDB: claves de idempotencia, outbox, locks.

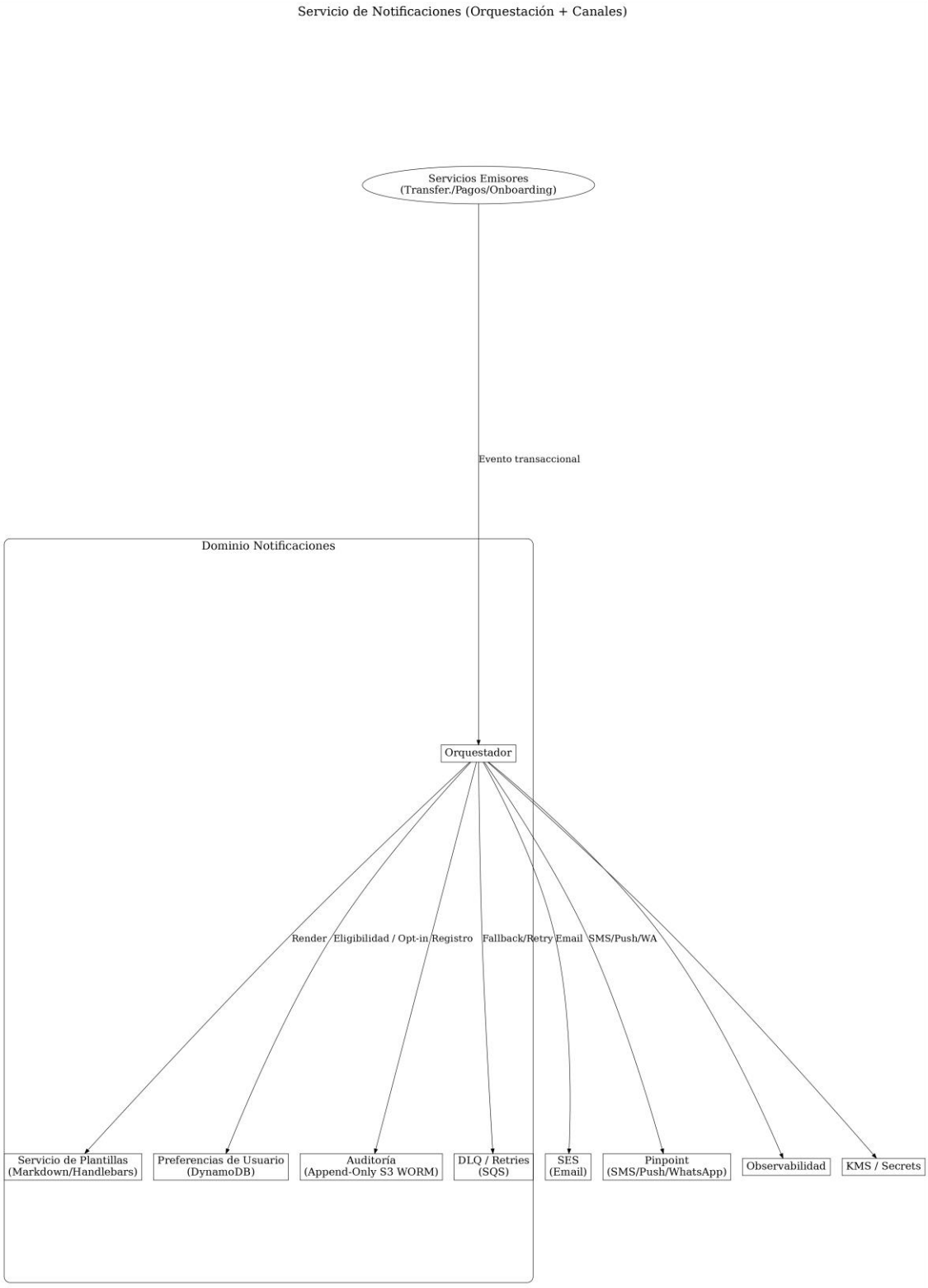
Secuencia (simplificada)

1. BFF valida JWT y envía orden al orquestador.
2. Orquestador → Antifraude/Límites → Idempotencia.
3. Reserva en ledger (Aurora) y ejecución en Core; si es interbancaria, llamado a BCE.
4. Publica evento (EventBridge/SQS); confirma a UI (sincrónico o estado en cola).
5. Si falla un paso, ejecuta compensación (reverso de reserva) y notifica.

Decisiones (con 2+ alternativas)

- Aurora vs event sourcing completo (Kafka + snapshots) → Se elige Aurora por simplicidad inicial y ACID; se deja event sourcing para evolución si se requieren auditorías de cambio a nivel evento.
- Antifraude interno vs proveedor externo → POV interno (reglas/umbrales); producción puede integrar un motor externo especializado si el banco lo requiere.

8.3 Servicio de Notificaciones (Orquestación + Canales)



Responsabilidades

- Orquestar envíos multi-canal (email, SMS, push, WhatsApp).
- Aplicar preferencias (opt-in/opt-out, quiet hours) y políticas regulatorias.
- Plantillado, localización y A/B.
- Persistir traza de entrega para auditoría y troubleshooting.

Patrones

- Outbox + DLQ; idempotencia por messageId.

Datos

- DynamoDB: preferencias por usuario/canal.
- S3 (WORM): copia de notificaciones relevantes con retención.

Decisiones

- Pinpoint como hub de SMS/Push/WA vs múltiples proveedores → Se elige Pinpoint por centralización; alternativa: Twilio/otros si cobertura local lo exige.
- Plantillas en servicio propio vs Plantillas Pinpoint → Se prefiere servicio propio (portabilidad y versionado), con sincronización a Pinpoint si hace falta.

8.4 Servicio de Auditoría (Append-Only / No-repudio)

Responsabilidades

- Registrar todas las acciones relevantes con metadatos (actor, dispositivo, IP, traceId).
- Garantizar integridad (hash encadenado) y retención según política.
- Exponer consultas para forense y cumplimiento.

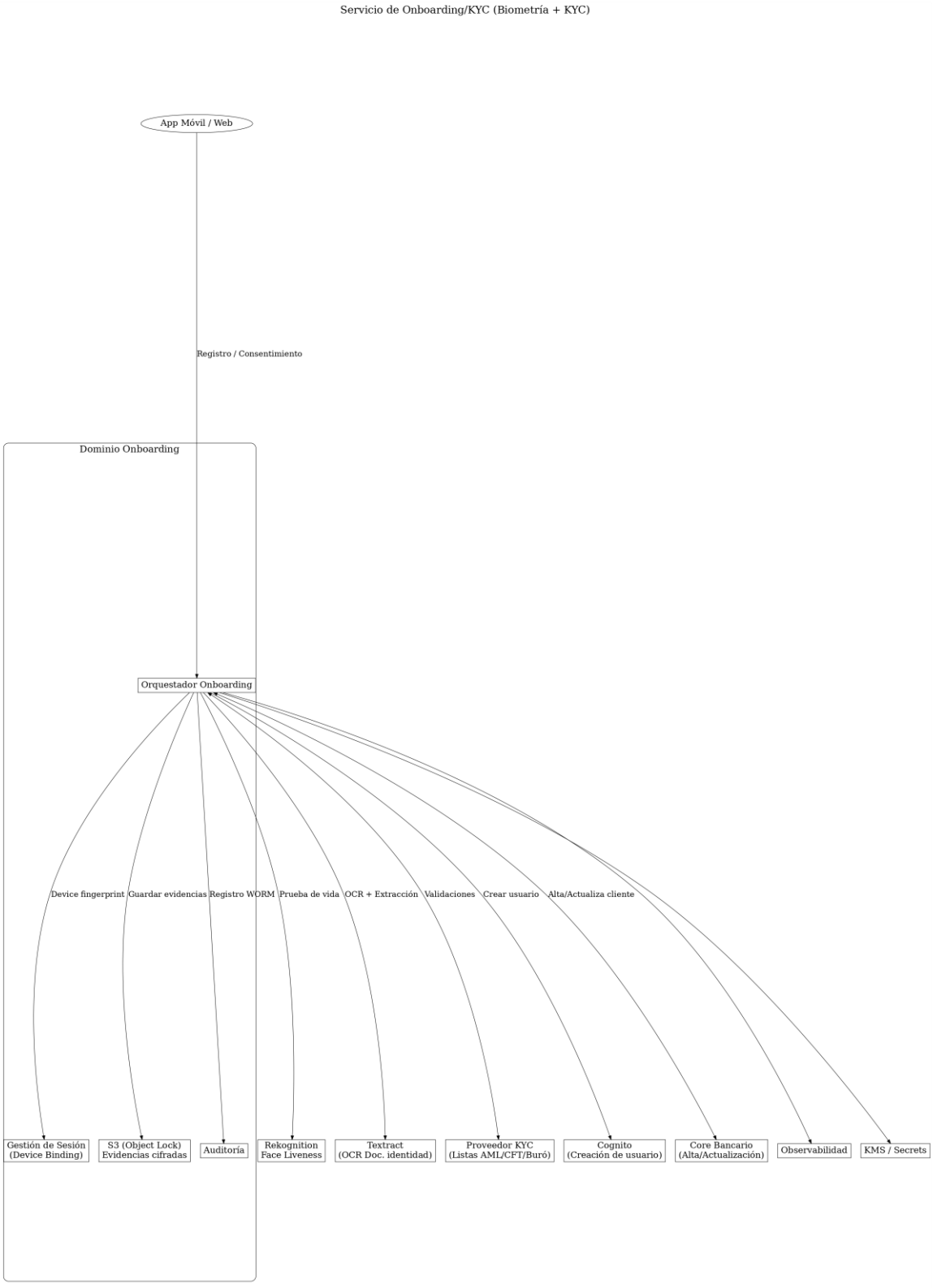
Almacenamiento

- S3 + Object Lock (WORM) para inmutabilidad.
- Opcional: QLDB para verificabilidad criptográfica; Athena para consulta ad-hoc.

Decisiones

- S3 WORM vs QLDB como repositorio primario → Se elige S3 WORM por costo y simplicidad; QLDB como complemento cuando se requiera verificación en cadena.

8.5 Servicio de Onboarding/KYC (Biometría + KYC)



Responsabilidades

- Captura biométrica con prueba de vida, OCR del documento y validaciones contra listas AML/CFT/buró.
- Gestión de consentimientos y retención de evidencias.
- Alta de usuario en Cognito y actualización/creación en Core.

Flujo (resumen)

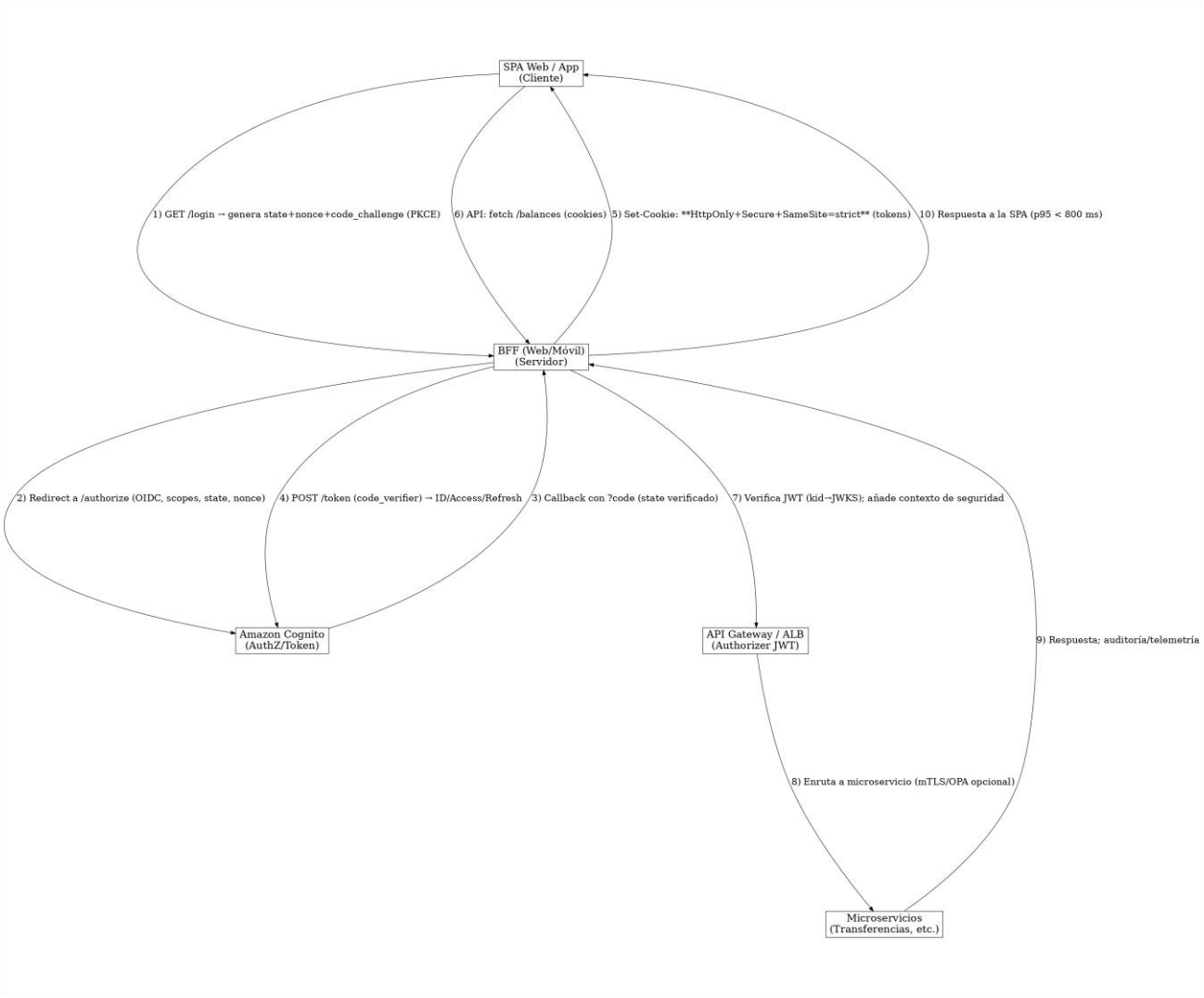
1. App pide consentimiento → captura selfie y documento.
2. Liveness (Rekognition) + OCR (Textract).
3. Validación con KYC; si aprueba, alta en Cognito y Core.
4. Evidencias cifradas en S3 (Object Lock); registro en auditoría.

Decisiones

- Rekognition vs proveedor externo (FaceTec/Onfido). Se inicia con Rekognition; si se requiere certificación específica o cobertura fuera de AWS, se integra externo vía adapter.
- Guardar plantillas biométricas vs hash/descriptor mínimo. Se elige mínima retención conforme LOPDP (solo lo estrictamente necesario y con caducidad definida).

9. Autenticación y autorización (OAuth2.0/OIDC)

Objetivo: proveer identidad confiable para **clientes** (web/móvil) y **colaboradores** (backoffice) con estándares abiertos, cumplimiento **LOPDP** y controles bancarios.



9.1 Patrón recomendado (clientes)

- **Authorization Code + PKCE** para SPA y móvil.
- **BFF** (Backend for Frontend) canjea el **code** por tokens y los entrega como **cookies HttpOnly + Secure + SameSite=strict** (evita exponer tokens a JS/localStorage).
- **API Gateway/ALB** con **Authorizer JWT** de Cognito; microservicios validan aud, iss, exp, jti, nonce y aplican controles de autorización.

9.2 Proveedor de Identidad (IdP)

- **Amazon Cognito – User Pools** como IdP de clientes.
- **Federación** con Microsoft Entra ID / Google / Apple vía **OIDC/SAML** (opcional para social/corporativo).
- **IAM Identity Center (ex SSO)** para **colaboradores** (acceso a consolas, backoffice y herramientas internas), federado con Entra ID si corresponde.

9.3 Tokens y sesiones

- **Algoritmo:** RS256; validar kid contra **JWKS** de Cognito; cachear JWKS.
- **Vigencias sugeridas (producción):**
  - **ID token:** 5–10 min (UI claims).
  - **Access token:** 5–10 min (acceso a APIs).
  - **Refresh token:** 30–60 días con **rotación** y **detección de reuso**.
- **Web:** cookies HttpOnly; **CSRF** (token doble-envío o SameSite) y cabeceras X-Requested-With.
- **Móvil:** almacenamiento seguro (Keychain/Keystore), sin secretos embebidos.

9.4 MFA y autenticación adaptativa (step-up)

- **MFA**: TOTP/Push/SMS; preferir **WebAuthn/Passkeys** donde sea posible.
- **Step-up** por **riesgo/importe/acción** (ej. transferencias > umbral): exigir MFA o re-autenticación.
- **Claims** acr/amr para evidenciar nivel de autenticación.

9.5 Autorización (RBAC/ABAC)

- **RBAC** por roles (cliente, pyme, premium) + **ABAC** por atributos (límite, geolocalización, dispositivo confiable).
- **Scopes**: accounts:read, transfers:write, notifications:manage, etc.
- **Fine-grained**: considerar **Amazon Verified Permissions (Cedar)** u **OPA** (en EKS) para políticas contextuales.

9.6 Seguridad del Frontend

- **CSP** estricta, **SRI** para recursos, deshabilitar eval, X-Frame-Options: DENY, **HSTS**.
- Evitar localStorage/sessionStorage para tokens.
- **CORS** mínimo necesario; pinning TLS en móvil (App Attest / Play Integrity opcional).

9.7 Seguridad del Backend

- **JWT Authorizer** en API Gateway o **OIDC en ALB** (para apps HTTP).

- **mTLS** interno entre servicios; **rate-limit** por sub/IP; **anti-replay** con jti.
- **Rotación** de claves/secretos con **KMS/Secrets Manager**.

9.8 Backoffice (Workforce)

- Acceso por **IAM Identity Center** federado con Entra ID; **MFA** corporativo; **segregación de funciones** y registros de auditoría.

9.9 Checklist de configuración en AWS (resumido)

1. **Cognito User Pool** + **Custom Domain**.
2. **App clients** (Web sin secret, Móvil sin secret) con **Authorization Code + PKCE**.
3. **Federación** OIDC/SAML a Microsoft/Google/Apple (opcional).
4. **MFA** activado y políticas de riesgo/step-up.
5. **Resource Server** y **custom scopes** (accounts:read, transfers:write, ...).
6. **Lambda triggers** (pre/post authentication) para enriquecer claims (mínimo necesario por LOPDP).
7. **Authorizers** en **API Gateway** / **ALB OIDC**; JWKS cache.
8. **Cookies** HttpOnly/SameSite=strict en BFF; **CSRF** en endpoints sensibles.
9. **Logs** de Cognito a CloudWatch y métricas/alertas; CloudTrail habilitado.

9.10 Parámetros sugeridos

- **Bloqueo de cuenta** tras N intentos fallidos con **exponencial backoff**.
- **Rotación** de refresh token por uso; invalidación en fuga de dispositivo.
- **Tiempo de inactividad** de sesión: cierre a los 15–20 min (web) con aviso.

9.11 Cumplimiento LOPDP (Ecuador)

- **Base legal y consentimiento** explícito para biometría.
- **Minimización de datos** en tokens (evitar PII innecesaria).
- **Retención** y borrado programado de datos de autenticación.
- **Registro de consentimientos** y accesos para auditoría.

9.12 Pruebas y evidencias

- **Pentest** de flujos OIDC y MFA.
- **Revisión de configuraciones** (CIS, ASVS) y **simulaciones** de reuso de refresh tokens.
- **Drills** de revocación masiva (compromiso de IdP) y recuperación de servicio.

10. Onboarding con reconocimiento facial

Objetivo: habilitar un alta 100% digital y segura, reduciendo fraude de identidad y cumpliendo la LOPDP (Ecuador) y regulaciones bancarias (SB, UAFE para AML/CFT).

10.1 Alcance y principios

- Captura biométrica con prueba de vida (liveness) + OCR de documento.
- KYC/AML: validación contra listas (sanciones, PEP, UAFE), buró/crédito (si aplica).
- Consentimiento explícito y minimización de datos biométricos (solo lo estrictamente necesario, con retención definida).
- Privacidad por diseño y seguridad por diseño (cifrado E2E, acceso mínimo, auditoría).

10.2 Flujo propuesto (resumen)

1. Consentimiento LOPDP en la app (propósito, retención, derechos ARCO, transferencias internacionales si aplica).
2. Captura selfie y liveness (detección spoof/deepfake; calidad de imagen mínima, iluminación, enfoque).
3. Captura documento (frontal/reverso), OCR y validaciones básicas (MRZ, fecha, formato).
4. KYC/AML con proveedor externo (listas, coincidencias, buró); scoring de riesgo.
5. Decisión: aprobado → creación de usuario en Cognito y alta/actualización en Core; dudoso → revisión manual.
6. Evidencias mínimas a S3 (Object Lock, KMS) y auditoría; notificación al cliente del resultado.

10.3 Arquitectura en AWS (POV → Producción)

- Orquestación: AWS Step Functions (o Lambda orquestador) con estados, timeouts y rutas de error.
- Biometría: Amazon Rekognition Face Liveness; umbral configurable; soporte anti-spoof.
- OCR: Amazon Textract (documentos de identidad). Validaciones de formato/consistencia.
- KYC/AML: Integración con proveedor (API REST/Async). Reintentos con SQS + DLQ.
- Almacenamiento de evidencias: S3 con Object Lock (WORM), KMS CMK, bloqueo de acceso público. Políticas de retención.
- Identidad: Cognito (User Pool). MFA sugerida post-alta.
- Core: Adapter ACL; idempotencia DynamoDB para evitar altas duplicadas.
- Observabilidad: CloudWatch/X-Ray, trazas y métricas; alarmas ante tasas anómalas de rechazo o latencia.

10.4 Tratamiento legal (LOPDP – Ecuador)

- Base legal: consentimiento explícito para biometría; informar finalidad, plazos, y derecho a revocar.
- Minimización: almacenar resultado de liveness y metadatos; evitar guardar plantillas completas salvo necesidad documentada.
- Retención: definir política (ej. 1–3 años para evidencias de onboarding, según criterio legal/riesgos). Borrado seguro tras vencimiento.
- Transferencias internacionales: verificar país/encargado con nivel adecuado o usar cláusulas contractuales; registrar evaluación.
- Derechos ARCO: mecanismo para acceso/rectificación/supresión; trazabilidad en auditoría.

10.5 Métricas/umbrales de calidad

- Tasa de finalización del onboarding ≥ 85% (POV), ≥ 90% (Producción).
- Liveness: FAR (False Accept Rate) ≤ 0.1%, FRR (False Reject Rate) ≤ 3% (inicial; ajustar por pruebas).
- Tiempo total (p95) ≤ 120 s con red 4G promedio.
- Revisión manual: ≤ 5% de casos en producción.
- Tiempo de decisión: automático ≤ 10 s; con revisión ≤ 4 h.

10.6 Riesgos y mitigaciones

- Spoofing/deepfakes → liveness activo/pasivo, detección de manipulación, calidad mínima de cámara.
- Fuga de PII/biometría → cifrado KMS, acceso mínimo IAM, S3 bloqueado, auditoría, VPC endpoints.
- Sesión robada → device binding, tokens efímeros, expiración corta, detección de anomalías.
- Sesgos biométricos → pruebas con muestra representativa local; monitoreo de FRR por cohortes; ajuste de umbrales.
- Conectividad limitada → captura offline-temporal y reintento; tamaño de media comprimido.

10.7 Almacenamiento y retención de biometría

- Evidencias (imágenes y resultados) en S3 con Object Lock, cifradas con KMS CMK; acceso mediante roles de mínimo privilegio.
- Embeddings: si se usan para deduplicación, almacenar hash/descriptor salado con pepper en Secrets Manager; rotación periódica.
- Borrado: jobs programados (Lifecycle + Lambda) y evidencia de eliminación.

10.8 Alternativas evaluadas

- Rekognition Face Liveness vs proveedores externos (Onfido/FaceTec):
  - Rekognition: integración nativa en AWS, latencia baja; costo por uso.
  - Externo: certificaciones/reglas locales, SDKs avanzados; costo/licenciamiento mayor; dependencia del proveedor.
- OCR Textract vs SDK de terceros: Textract reduce dependencias; terceros pueden tener mejores modelos para documentos locales.

10.9 Pruebas, UAT y cumplimiento

- Datasets locales (variabilidad de piel, iluminación, dispositivos).
- Calibración de umbrales FAR/FRR y tiempos.
- PenTest y privacy impact assessment (PIA/DPIA) documentados.
- Ensayos de carga (picos de campañas) y drills de caídas de proveedor KYC con fallback a revisión manual.

11. Integración con Core y Sistema Complementario (AWS)

Objetivo: integrar la Banca por Internet (BP) con Core Bancario, BCE y proveedores KYC/Complementarios con bajo acoplamiento, resiliencia, idempotencia y trazabilidad, usando servicios nativos de AWS.

11.1 Patrón de integración y Anti-Corrupción (ACL)

- Se usará un Servicio Adaptador (ACL) por dominio (Cuentas, Movimientos, Transferencias, Pagos) desplegado en Amazon EKS.
- Justificación: encapsula transformaciones (SOAP/REST→REST), validaciones y reglas de negocio de borde, evitando que la complejidad del Core impacte a los microservicios; facilita pruebas y versionado independiente.

11.2 Conectividad segura al Core y sistemas on-prem

- Se usará AWS Direct Connect con AWS Transit Gateway para conectividad dedicada entre VPC y datacenter del banco; en POV se mantiene VPN pero el objetivo de producción es Direct Connect.
- Se usará Network Load Balancer (NLB) con targets por IP para enrutar llamadas de EKS hacia servicios del Core a través de Direct Connect.
- Se usará TLS extremo a extremo y ACM Private CA para certificados internos.
- Justificación: latencia estable, rutas controladas, cifrado gestionado y aislamiento de red alineado a requisitos de la SB.

11.3 Integración síncrona (lecturas/órdenes)

- Se usará API Gateway (REST/HTTP) como entrada a BP y EKS para la lógica. Los Adaptadores ACL llaman al Core mediante NLB y Direct Connect.
- Se usará RDS Proxy frente a Aurora PostgreSQL para proteger el pool en picos.
- Justificación: limita superficies públicas, protege bases transaccionales y mantiene contratos consistentes hacia el front.

11.4 Integración asíncrona (eventos y compensaciones)

- Se usará Amazon SQS FIFO para eventos críticos de dinero (orden, confirmación, compensación) con DLQ y reintentos; Amazon EventBridge para eventos de negocio de difusión (ej. "transfer_completed").
- Se usará Transactional Outbox persistido en DynamoDB; un Lambda publisher publica a SQS FIFO/EventBridge con messageDeduplicationId y messageGroupId.
- Justificación: garantiza orden e idempotencia en dinero, y desacopla productores/consumidores con trazabilidad completa.

11.5 CQRS (consultas vs. comandos)

- Se usará CQRS: consultas desde BFF a proyecciones cacheadas (Redis) y comandos a servicios transaccionales en Aurora.
- Justificación: baja latencia para lectura intensiva sin sobrecargar Core ni bases de escritura.

11.6 Idempotencia y deduplicación

- Se usará una tabla DynamoDB idempotency con partición por operationId y TTL (24–48h). Los servicios validan y retornan el mismo resultado si la operación se reintenta.
- Se usará SQS FIFO con deduplicación de mensajes y claves de grupo por cuenta/cliente.
- Justificación: evita duplicados en transferencias/pagos y soporta reintentos seguros.

11.7 Resiliencia: timeouts, retries y circuit breaker

- Se usará timeouts definidos en AWS AppConfig por integración (Core, BCE, KYC) y retries con backoff exponencial (SDK + política de servicio).
- Se usará circuit breaker (resilience4j/Envoy) en los Adaptadores ACL; estado expuesto en métricas.
- Justificación: preserva capacidad del sistema frente a fallas aguas abajo y acota el consumo de error budget.

11.8 Contratos, versionado y descubrimiento

- Se usará OpenAPI para APIs REST de BP (gestionadas en API Gateway, con stages y deployments versionados).
- Se usará AsyncAPI para eventos y EventBridge Schema Registry para catalogarlos.
- Se usará Pact para pruebas de contrato en CI/CD.
- Justificación: cambios controlados, compatibilidad hacia atrás y descubrimiento centralizado de eventos.

11.9 Seguridad de integración

- Se usará IAM roles de mínimo privilegio para cada servicio; Secrets Manager para credenciales de terceros; KMS para cifrado de secretos y discos.
- Se usará Security Groups restrictivos por microservicio, NACLs y VPC endpoints para servicios AWS.
- Se usará mTLS interno en EKS (malla opcional) para llamadas servicio-a-servicio.
- Justificación: defensa en profundidad y trazabilidad de acceso por identidad técnica.

11.10 Observabilidad de la integración

- Se usará OpenTelemetry + AWS X-Ray para trazas end-to-end; CloudWatch para métricas/logs; Grafana para paneles.
- Se usará correlación por traceId/txnId desde BFF hasta Adaptadores/Core; alarmas por latencia, tasa de error, tamaño de colas y circuit breaker abierto.
- Justificación: diagnóstico rápido, SLOs medibles y auditoría técnica.

11.11 Flujos de referencia

1. Consulta de movimientos: BFF → Proyección (Redis); si falta, lectura a Aurora; si requiere dato en Core, Adaptador ACL → Core por Direct Connect; respuesta con traceId.
2. Transferencia propia: BFF → Servicio de Transferencias → idempotencia (DynamoDB) → asiento en Aurora → ejecución en Core vía ACL → evento transfer_completed en EventBridge → notificación.
3. Transferencia interbancaria: igual al anterior, agregando llamada a BCE; estado final puede ser asíncrono, informado por SQS/EventBridge.

11.12 Criterios de aceptación

- p95 de latencia para consultas ≤ 500 ms desde proyección; ≤ 1.2 s para transferencias propias; ≤ 2.5 s para interbancarias.

- 0 duplicados en operaciones monetarias confirmadas (idempotencia validada).
- 99.9% disponibilidad de la capa de integración en producción.
- Drills de caída de Core/BCE con degradación controlada y colas sin pérdida.

12. Notificaciones

Objetivo: entregar notificaciones transaccionales y operativas de manera segura, trazable y conforme a LOPDP en Ecuador, usando AWS donde corresponde y proveedores especializados cuando el canal lo requiere.

12.1 Canales y servicios elegidos

- Email: Amazon SES (Simple Email Service).
  - Justificación: servicio AWS nativo, costo-eficiente, alta entregabilidad con DKIM/SPF/DMARC gestionados y eventos de retroalimentación (bounces/complaints) para listas sanas.
- Push móvil: AWS SNS – Mobile Push hacia APNs (iOS) y FCM (Android).
  - Justificación: integración directa con proveedores de push, manejo de device tokens por plataforma y escalabilidad gestionada.
- SMS y WhatsApp: Twilio Programmable Messaging (incluye WhatsApp Business), con plantillas aprobadas por Meta cuando aplique.
  - Justificación: cobertura telco y canal WhatsApp robusto; simplifica cumplimiento de requisitos del canal y estados de entrega vía webhooks firmados.

12.2 Arquitectura de orquestación (en AWS)

- Servicio de Notificaciones en Amazon EKS (o Lambda si es serverless), expuesto detrás de API Gateway.
- Preferencias del usuario (opt-in/opt-out, quiet hours, idioma/canal preferido) en DynamoDB.
- Plantillas (multi-idioma, variables, versiones) en S3 con versionado; render en el servicio (Markdown/Handlebars) y copia WORM para auditoría cuando aplique.
- Colas SQS (FIFO) para encolar solicitudes y asegurar orden/idempotencia por usuario; DLQ para fallos persistentes.
- Eventos de negocio desde EventBridge (p.ej., transfer.completed, payment.posted).
- Integraciones de salida:
  - SES (SDK AWS) para email.
  - SNS Mobile Push (APNs/FCM) para push.
  - Twilio (HTTPS) para SMS/WhatsApp con webhooks de estado.
- Secretos y claves API en AWS Secrets Manager; cifrado con KMS.

- Observabilidad: OpenTelemetry + X-Ray, métricas en CloudWatch/Grafana (tasa de entrega, latencia, rebotes).

12.3 Flujo de envío (resumen)

1. Un servicio de dominio publica un evento (p.ej., transfer.completed).
2. EventBridge enruta al Servicio de Notificaciones.
3. El servicio consulta preferencias y selecciona canal (con quiet hours).
4. Renderiza la plantilla (datos mínimos necesarios) y encola en SQS FIFO con idempotency key.
5. Consumidor envía por SES/SNS/Twilio según canal.
6. Recibe callback/evento de entrega (SES/SNS/Twilio) → registra estado en Aurora/DynamoDB y escribe auditoría en S3 (WORM).

12.4 Conformidad LOPDP y políticas de canal

- Consentimiento y propósito: cada canal requiere opt-in explícito almacenado (con timestamp y fuente).
- Minimización: incluir solo los datos estrictamente necesarios en el mensaje (evitar PII sensible en SMS/WA).
- Retención: política de conservación de contenidos y metadatos (p.ej., 18–36 meses para auditoría), con borrado programado.
- Transferencias internacionales: evaluar ubicación de Twilio/Meta/SES; documentar medidas (cláusulas contractuales, DPA) y registrar evaluación de impacto.
- WhatsApp: usar plantillas aprobadas por Meta, con variables validadas; respetar ventanas de conversación.

12.5 Seguridad y anti-abuso

- Firma y validación de webhooks (Twilio/Meta) y Auth en endpoints (API Gateway + IAM/JWT).
- Rate limiting por usuario/IP/canal (Redis).
- Doble envío: prevenir con idempotency key y messageId único.
- CSP/DMARC: para email, configurar SPF/DKIM y DMARC con monitoreo de reportes.
- WAF + Shield en el perímetro; mTLS interno para llamadas entre servicios.

12.6 Métricas y SLO de canal

- Email (SES): tasa de entrega ≥ 98%, rebotes < 2%, quejas < 0.1%.
- Push (APNs/FCM): latencia p95 < 1s desde enqueue; tokens inválidos < 1% (limpieza periódica).
- SMS/WhatsApp (Twilio): D+ (delivered) ≥ 95% en rutas principales; latencia p95 < 3s; tracking de messageSid.
- Global: correlación por traceId end-to-end y panel de entregabilidad por canal.

12.7 Checklist de implementación

- SES: dominio verificado, DKIM/SPF/DMARC activos; configuration sets para eventos de entrega.
- SNS Push: crear platform applications (APNs/FCM), gestionar tokens y limpieza.
- Twilio: credenciales en Secrets Manager, webhooks con validación de firma y reintentos; plantillas de WhatsApp registradas y aprobadas.
- Servicio: versionado de plantillas, preferencias/consentimientos en DynamoDB, SQS FIFO + DLQ, S3 (WORM) para auditoría, alertas por caída de entregabilidad.

13. Persistencia y datos

Objetivo: garantizar consistencia transaccional, baja latencia de lectura, trazabilidad/auditoría y cumplimiento LOPDP con una estrategia de datos clara y segura.

13.1 Fuente de verdad transaccional

- Se usará: Amazon Aurora PostgreSQL.
  - Producción: Multi-AZ con Read Replicas y RDS Proxy para manejo de conexiones.
  - POV/MVP: Aurora Serverless v2 (mismo engine, escala granular).
  - Justificación: ACID para dinero, alto rendimiento, HA nativa y compatibilidad SQL para procesos bancarios y conciliaciones.

Esquemas y modelado

- Modelo por dominio: customers, accounts, transactions, payments, devices, audit_links.
- Ledger transaccional: asientos con debe/haber, transaction_id global, control de idempotencia y versionado optimista (xmin o campo version).
- Índices clave: por account_id + date, customer_id, transaction_id; particiones por rango de fecha para tablas de movimientos.

Operación

- Backups automáticos y PITR; maintenance windows controladas; parameter groups endurecidos (TLS requerido, log_min_duration_statement).

13.2 Idempotencia, outbox y locks

- Se usará: Amazon DynamoDB (tabla idempotency) con TTL 24–48h y GSI por requestKey.

- Outbox: tabla outbox en DynamoDB consumida por Lambda publisher → SQS FIFO/EventBridge.
- Locks de cortísima duración por account_id/customer_id para operaciones sensibles.
- Justificación: velocidad O(1), escalabilidad y consistencia de aplicación para evitar duplicados en dinero.

13.3 Caché y rate-limit

- Se usará: Amazon ElastiCache for Redis (Multi-AZ con failover, TLS, AUTH).
    - Caché de consultas de lectura intensiva (saldos, últimos movimientos) con TTLs cortos.
    - Rate-limit por sub/IP/canal; tokens anti-replay (jti).
    - Justificación: latencias p95/p99 bajas sin cargar la base transaccional.

13.4 Documentos, evidencias y archivos

- Se usará: Amazon S3 con Object Lock (WORM) y KMS CMK.
    - Buckets segregados por dominio (evidences-kyc, statements, exports) con Block Public Access.
    - Lifecycle a Glacier para archivado; replicación cross-Region opcional para DR.
    - Justificación: retención inmutable para auditoría y costos óptimos a lo largo del ciclo de vida.

13.5 Búsqueda operativa y consultas ad-hoc

- Se usará: Amazon OpenSearch Service para búsquedas textuales y consultas operativas (p. ej., “por referencia”, “por mensaje de error”, trazas enriquecidas).
    - Ingesta desde EventBridge/SQS y de logs (a través de Kinesis Firehose si se requiere).
    - Justificación: respuestas rápidas en criterios no indexados relacionales y soporte a backoffice/Soporte.

13.6 Cifrado, secretos y control de acceso

- Cifrado en reposo: KMS para Aurora, DynamoDB, S3 y snapshots.
- Gestión de secretos: AWS Secrets Manager (rotación programada) y SSM Parameter Store para configuración.
- Accesos: IAM Roles de mínimo privilegio por servicio; RLS (Row-Level Security) en PostgreSQL para vistas de backoffice cuando aplique.

13.7 Calidad de datos y no repudio

- Auditoría de cambios críticos mediante servicio de Auditoría (eventos firmados, hash encadenado) con copia en S3 (WORM).
- Constraints y triggers en Aurora para integridad referencial; validaciones previas en capa de dominio.
- Trazabilidad por traceId/txnId desde la UI hasta la base.

13.8 Datos en no-producción (LOPDP)

- Se usará: datos sintéticos o enmascarados para QA/DEV.
- Automatización de desidentificación (scripts/ETL) antes de exportar; prohibido mover PII/biometría sin base legal.
- Justificación: cumplir LOPDP y reducir riesgo de exposición en ambientes no controlados.

13.9 Monitoreo y métricas de datos

- Aurora: conexiones activas, lag de réplica, locks, deadlocks, latencias p95.
- DynamoDB: RCU/WCU, throttling, latencia y tasa de ConditionalCheckFailed.
- Redis: evictions, hit ratio, memoria; SQS: tamaño de cola y age máximo.
- OpenSearch: CPU/heap, latencia de consulta y errores.

13.10 DR y copias

- Aurora: snapshots automáticos y réplicas; PITR.
- DynamoDB: PITR; backups programados por tabla.
- S3: versión + Glacier; validación periódica de restauración.

13.11 Criterios de aceptación

- Consistencia transaccional garantizada en Aurora; 0 duplicados por idempotencia.
- Latencia p95 de consultas más usadas ≤ 500 ms (desde caché) y ≤ 1.2 s (desde Aurora).
- Cumplimiento: cifrado KMS, acceso mínimo IAM, registros de auditoría verificables.
- Restauraciones de backup validadas trimestralmente.

14. Auditoría y no repudio

Objetivo: garantizar integridad, inmutabilidad y verificabilidad criptográfica de los registros de auditoría, habilitando cadena de custodia y consulta forense conforme a LOPDP y normativa de la Superintendencia de Bancos en Ecuador.

14.1 Qué se audita

- Autenticación (login/logout, MFA, step-up), consentimientos y cambios de preferencias.
- Operaciones de dinero (órdenes, validaciones, resultados), notificaciones (messageId/estado), administración de dispositivos.
- Accesos administrativos y cambios de configuración (AppConfig, Feature Flags, roles).
- Accesos a datos sensibles y consultas forenses (quién, cuándo, desde dónde).

14.2 Diseño de inmutabilidad y verificación

- Se usará un Servicio de Auditoría en Amazon EKS que recibe eventos en JSON con metadatos (actor, sub, ip, device, traceId, txnId, timestamp), calcula hash SHA-256 y firma con AWS KMS (clave asimétrica).
    - Justificación: cada evento queda firmado y ligado a una identidad técnica y a un sello de tiempo confiable.
- Se usará encadenamiento (prevHash → chainHash) por particiones temporales (por hora/día) para formar una cadena tamper-evident.
    - Justificación: si se altera un evento, se rompe la cadena y la verificación falla.
- Se usará Amazon QLDB para anclar el digest (raíz de la cadena) a intervalos (horario/diario) y exponer pruebas criptográficas de integridad.
    - Justificación: provee verificabilidad independiente y rápida de que un evento en S3 no ha sido alterado.

14.3 Consulta forense y reportes

- Se usará Athena + Glue Catalog sobre los datos en S3 (particionados por fecha/servicio/acción) para búsquedas y reportes de auditoría.
    - Justificación: consultas ad-hoc sin mover datos, con costo por escaneo y performance aceptable.
- Se usará una API de Verificación que toma un evento de S3, recomputa su hash y lo contrasta con el digest anclado en QLDB; genera un informe firmado (PDF/JSON) de integridad.
    - Justificación: facilita entregables regulatorios y peritajes.

14.4 Seguridad, accesos y segregación de funciones

- Se usará IAM de mínimo privilegio; roles sólo-lectura forense separados de roles de operación.
- Se usará AWS Secrets Manager para claves de webhooks/proveedores y KMS para llaves de firma.
- Se usará CloudWatch Logs para la operación del servicio de auditoría y CloudTrail para registrar accesos a S3/KMS/QLDB.
    - Justificación: segregación de funciones y trazabilidad de cada acceso a evidencias.

14.5 Retención, LOPDP y cadena de custodia

- Retención definida por política (p.ej., 5–7 años para registros transaccionales; 1–3 años para onboarding), con Lifecycle a Glacier y borrado programado.
- PII/biometría: almacenar mínimos metadatos necesarios; para evidencias sensibles, guardar enlaces y hashes (no contenido completo) cuando esté permitido.
- Cadena de custodia: metadatos who/when/where/how; firmas KMS; export de logs de acceso; informes de verificación.

14.6 Monitoreo y SLO

- Alarmas por fallas de entrega (Firehose), expiración de llaves KMS, intentos de borrado/modificación en S3 con Object Lock, y divergencias en verificación.
- SLO: ingestión ≥ 99.99%, latencia de verificación ≤ 2 s por evento (p95), disponibilidad del repositorio ≥ 11×9 (S3).

14.7 Criterios de aceptación

- Demostración de inmutabilidad: intento de alteración detectado por cadena de hashes y QLDB.
- Consulta forense sobre un período y servicio con Athena en < 5 min.
- Evidencia de retención/borrado conforme a política y LOPDP.
- Informe de verificación generado para un conjunto de eventos y validado por auditoría interna.

15. Alta disponibilidad, resiliencia y DR

Objetivo: mantener el servicio operativo ante fallas de zona/región y recuperar sin pérdida material de datos, con metas de RTO ≤ 30 min y RPO ≤ 5 min en escenario multi-región, y alta disponibilidad intra-región para incidentes comunes.

15.1 Disponibilidad intra-región (Primaria)

- Se usará despliegue Multi-AZ en la región primaria.
  - EKS con 3 zonas y PodDisruptionBudgets, anti-affinity, HPA y liveness/readiness.
  - Aurora PostgreSQL Multi-AZ con RDS Proxy para estabilidad de conexiones.
  - DynamoDB (HA regional por diseño) y ElastiCache Redis con réplica y failover automático.
  - ALB/API Gateway en subredes de múltiples AZ, CloudFront + WAF/Shield en el perímetro.
  - Justificación: elimina puntos únicos de falla dentro de la región y permite mantenimiento sin downtime.

15.2 Recuperación ante desastre (DR) multi-región — Warm Standby

- Se usará topología primaria ↔ secundaria en modo warm standby para balancear costo/tiempo de recuperación.
  - Aurora Global Database con réplica read-only en la secundaria; promoción a writer durante conmutación.
  - DynamoDB Global Tables para replicación multi-región de idempotencia/outbox/locks.
  - S3 con CRR (Cross-Region Replication) y Object Lock (WORM) para evidencias.
  - EventBridge Global Endpoint para failover automático de publicación/consumo de eventos.
  - SQS aprovisionado en standby; activación al conmutar.
  - OpenSearch con Cross-Cluster Replication para búsquedas/logs operativos.
  - KMS Multi-Region Keys (MRK) y Secrets Manager sincronizado por pipeline.
  - EKS mínimo en secundaria con BFF + servicios críticos, capacidad autoscalable.
  - Cognito pre-provisionado en secundaria con sincronización de usuarios por triggers (replicación de altas/cambios) para continuidad de login.
  - Justificación: garantiza RTO/RPO definidos con costos controlados al mantener capacidad mínima activa y datos replicados.

15.3 Conmutación y retorno (Runbooks)

- Se usará Route 53 con Health Checks y política Failover/Latency-based para dirigir tráfico.
- Pasos de conmutación (resumen):
    1. Congelar escrituras no críticas; verificar salud de réplicas.
    2. Promover Aurora Global a writer en secundaria.
    3. Aumentar capacidad EKS secundaria; calentar caché Redis.
    4. Habilitar colas SQS y consumidores en secundaria.
    5. Cambiar orígenes de CloudFront/ALB y conmutar DNS en Route 53 (TTL bajo).
    6. Monitorear error-rate/latencias; activar runbooks de contingencia.
- Retorno (failback): validación de coherencia, re-promoción a primaria, reconfiguración de orígenes y gradual desvío de tráfico.
- Justificación: procedimiento repetible y auditable que minimiza el tiempo fuera de servicio.

15.4 Consistencia de datos y reconciliación

- Se usará idempotencia por operationId en DynamoDB y Transactional Outbox para garantizar exactly-once lógico.
- Tras la conmutación, un job de reconciliación re-procesa DLQ y verifica ledger (Aurora) vs eventos aplicados.
- Sesiones: invalidación selectiva y re-autenticación si corresponde; tokens con expiración corta.
- Justificación: evita dobles cargos o pérdidas de eventos durante y después del failover.

15.5 Observabilidad y pruebas

- Synthetics (canarios) por canal (login, consulta, transferencia) en múltiples regiones.
- SLO/Burn rate con alertas; paneles por región y trazas OTel/X-Ray entre regiones.
- Game Days/Chaos: derribo de AZ, pérdida de conectividad a Core/BCE, caída forzada de writer Aurora, simulación de saturación de colas.
- Justificación: evidencia de confiabilidad real y tiempos de detección/recuperación medibles.

15.6 Controles de seguridad en DR

- WAF/Shield activos en ambas regiones; mTLS interno; IAM con roles separados por región.
- CloudTrail/Config multi-región y Security Hub/GuardDuty habilitados; copias de logs en cuenta de seguridad.
- Justificación: continuidad de postura de seguridad durante y después del failover.

15.7 Costos operativos

- Warm standby mantiene recursos mínimos en secundaria (EKS reducido, colas standby, réplicas de datos) y escala on-demand al conmutar.
- Aurora Global y Global Tables incurren en costo de replicación; se monitorea con AWS Budgets y etiquetas FinOps.
- Justificación: equilibrio entre resiliencia fuerte y gasto sostenible.

15.8 Criterios de aceptación

- RTO ≤ 30 min y RPO ≤ 5 min verificados en drills semestrales con informe.
- P95 de latencia post-conmutación dentro de objetivos en < 15 min desde la activación.
- 0 duplicados en operaciones monetarias confirmadas tras reconciliación.
- Evidencias de ejecución de runbooks y auditoría de acciones (CloudTrail) disponibles.

16. Observabilidad y monitoreo

Objetivo: detectar y resolver incidentes antes de que impacten, con telemetría end-to-end (métricas, logs, trazas, canarios), SLOs claros y alertas accionables, cumpliendo LOPDP (no PII en logs) y buenas prácticas bancarias.

16.1 Instrumentación estándar

- Se usará OpenTelemetry (OTel) en servicios (SDK + ADOT Collector) para métricas y trazas.
- Trazabilidad: W3C trace-context (traceId/spanId) propagado desde el BFF hasta cada microservicio y a llamadas a Core/BCE; el BFF genera txnId (negocio) para correlación.
- Logs estructurados (JSON) en todos los componentes con campos mínimos: timestamp, severity, service, env, region, traceId, spanId, txnId, userHash, ipHash, error.code, message.
- Política de datos: no registrar credenciales, tokens, PII ni biometría; mascar PAN/cuentas (sólo 4 últimos), ofuscar IP/dispositivo (hash salado).
- Muestreo de trazas: 10% base; 100% al detectar errores o latencia > p95.

16.2 Plataforma de observabilidad en AWS

- Métricas: Amazon CloudWatch Metrics + Amazon Managed Service for Prometheus (AMP) para *-exporter en EKS (kube-state, node, app).
- Dashboards: Amazon Managed Grafana (carpetas por dominio/ambiente, control SSO por IAM Identity Center).
- Trazas: AWS X-Ray como backend de trazas (ingesta OTel → X-Ray); vistas por servicio, mapa de dependencias y análisis de latencia/errores.

- Logs: CloudWatch Logs con retención (p.ej., 180 días hot), export periódico a S3 (cuenta de seguridad) con KMS y lifecycle a Glacier (1–3 años). Logs Insights para consultas ad-hoc.
- Canarios: CloudWatch Synthetics para flujos críticos (login, saldo, transferencia, notificación), multi-región y con captura de evidencia.
- Frontend Web: CloudWatch RUM para medir LCP/CLS/INP, errores JS y latencia real.
- Alertado: CloudWatch Alarms → Amazon SNS → canales de Slack/Teams/email (vía webhook) y rotas de guardia.
- Seguridad (telemetría complementaria): CloudTrail, Security Hub, GuardDuty y Inspector integrados con cuentas de seguridad.

16.3 Dashboards y SLOs (por servicio y global)

- SLOs de canal (de NFR):
  - Login p95 ≤ 800 ms, Saldos p95 ≤ 500 ms, Transferencia propia p95 ≤ 1.2 s, Interbancaria p95 ≤ 2.5 s.
  - Disponibilidad canal ≥ 99.9% mensual; entregabilidad email ≥ 98%; WhatsApp/SMS ≥ 95%.
- Dashboards Grafana:
  - Golden Signals por servicio: latencia, error-rate (5xx/4xx), tráfico (RPS), saturación (CPU/memoria, conexiones DB, queue length SQS, throttles DynamoDB), health por zona.
  - Dependencias externas (Core/BCE/KYC): latencia, tasa de timeouts/circuit breaker, SLI ext.
  - DB (Aurora): lag de réplica, locks, deadlocks, IOPS; Redis: hit ratio, evictions; DynamoDB: WCU/RCU y throttling.
  - Negocio: transfers.success_ratio, avg_amount, tasa de fraude bloqueado, canales de notificación.

16.4 Alertas (accionables, sin ruido)

- Burn-rate SLO multi-ventana (ej.: 2% en 5 min y 1% en 1 h) para disponibilidad y error-rate.
- Latencia p95 por endpoint/servicio por encima de umbral (>20% del objetivo) sostenida 10 min.
- Backlogs: SQS ApproxAgeOfOldestMessage > 60 s o longitud > N (por servicio).
- Aurora: lag de réplica > 2 s; Redis: evictions > 0; DynamoDB: throttling > 0.5%.
- Canarios fallidos 2/3 iteraciones; RUM: LCP > 2.5 s en >10% sesiones.
- Rutas: severidad P1 (pager/on-call), P2 (chat + ticket), P3 (sólo ticket). Dedupe y silencios programados para mantenimientos.

16.5 Operación: runbooks e incidentes

- Runbooks por alerta: pasos de diagnóstico (dashboards/queries X-Ray/Logs), acciones de mitigación, criterios de cierre y comunicación.
- Game Days/Chaos trimestrales (derribo de AZ, caída de Core/BCE, failover Aurora, saturación de colas) con learnings documentados.
- Postmortems sin culpables; acciones y due-date rastreados (ticketing) y verificados.

16.6 Cumplimiento y gobierno de observabilidad

- Cifrado en tránsito y en reposo (KMS) para métricas/logs/trazas.
- Control de acceso por roles (sólo lectura vs admin), separación por cuenta de seguridad para almacenamiento largo plazo.
- Data hygiene: validadores en Collector para redactar PII/secretos; políticas de retención documentadas y auditables.
- Etiquetado (env, service, cost-center) para coste y responsabilidad.

16.7 Criterios de aceptación

- 100% de microservicios con OTel y correlación traceId/txnId activa.
- Dashboards por servicio + NOC global publicados en Grafana y referenciados en runbooks.
- Canarios en 2+ regiones por flujo crítico con SLO ≥ 99% éxito mensual.
- Alertas de SLO y recursos desplegadas con Infra as Code (Terraform/CDK) y probadas.
- Drill de incidente simulado resuelto dentro de MTTR objetivo ≤ 15 min.

17. Seguridad y cumplimiento normativo

Objetivo: proteger confidencialidad, integridad, disponibilidad y privacidad de la información del banco y de los clientes, cumpliendo LOPDP (Ecuador), lineamientos de la Superintendencia de Bancos, disposiciones del BCE y estándares internacionales (ISO/IEC 27001, PCI DSS si aplica tarjetas), implementados con servicios nativos de AWS.

17.1 Gobierno de seguridad y segregación de funciones

- Se usará una cuenta de seguridad dedicada (multi-cuenta AWS) para centralizar logs, auditoría y detección.
- Se usará IAM Identity Center para colaboradores, con MFA obligatorio, SSO, y roles diferenciados (operación, seguridad, auditoría, desarrollo).
- Se usará SCPs (Service Control Policies) para impedir acciones de alto riesgo a nivel organización (p. ej. desactivar CloudTrail, borrar buckets de evidencias, cambiar políticas KMS).
- Se usará un flujo break-glass (rol de emergencia) con aprobación fuera de banda, expiración y auditoría reforzada.
- Justificación: separación clara de responsabilidades y prevención de cambios no autorizados.

17.2 Gestión de identidades y accesos (principio de mínimo privilegio)

- Se usará IAM Roles por servicio con políticas least-privilege y permission boundaries para controlar lo que los equipos pueden delegar.
- Se usará IRSA (IAM Roles for Service Accounts) en EKS para que los pods asuman roles IAM sin credenciales estáticas.
- Se usará expiración corta de credenciales, rotación automática en Secrets Manager, y sesiones con duraciones máximas.
- Justificación: elimina credenciales largas, reduce superficie y permite trazabilidad por identidad técnica.

17.3 Protección de datos personales (LOPDP)

- Se usará minimización y propósito: sólo recolectar y procesar lo necesario (especial cuidado con biometría).
- Se usará consentimiento explícito para datos sensibles (biometría), con registro y revocación disponibles.
- Se usará retención definida y borrado programado (S3 Lifecycle/Lambda) de datos y evidencias.
- Se usará registro de solicitudes ARCO (acceso, rectificación, supresión, oposición, portabilidad) y su atención.
- Se usará evaluación y registro de transferencias internacionales de datos cuando apliquen.
- Justificación: alineado a LOPDP y mejores prácticas de privacidad por diseño.

17.4 Criptografía y gestión de claves

- Se usará cifrado en tránsito TLS 1.2+ extremo a extremo (ALB/API GW, mTLS interno opcional).
- Se usará cifrado en reposo con AWS KMS para Aurora, DynamoDB, S3, EBS, OpenSearch, Secrets y snapshots.

- Se usará KMS Multi-Region Keys (MRK) para llaves compartidas entre regiones (escenario DR) y políticas de rotación anual.
- Se usará AWS Certificate Manager (ACM) para certificados públicos; ACM Private CA para identidad interna (mTLS).
- Justificación: criptografía gestionada, auditada y con alta disponibilidad.

17.5 Seguridad de red y perímetro

- Se usará VPC por ambiente con subredes privadas para cómputo y datos; NACLs y Security Groups restrictivos (deny-all por defecto).
- Se usará VPC Endpoints (Gateway/Interface) para acceso privado a servicios AWS (S3, DynamoDB, Secrets, KMS, etc.).
- Se usará AWS WAF + Shield Advanced delante de CloudFront/ALB, con Bot Control y rate-limiting.
- Se usará PrivateLink para integraciones con terceros que lo soporten; Direct Connect para Core/BCE.
- Justificación: defensa en profundidad, sin exponer datos/servicios a Internet salvo lo estrictamente necesario.

17.6 Endurecimiento de cómputo (EKS/containers/Lambda)

- Se usará EKS con Pod Security Admission (perfil restricted), NetworkPolicies y namespaces por dominio.
- Se usará Amazon Inspector para escaneo de ECR (imágenes), Lambda y EC2; políticas para bloquear despliegues con CVEs críticos.
- Se usará OPA/Gatekeeper para políticas (no root, FS read-only, no privilegios, no hostPath, recursos limitados).
- Se usará IRSA, secrets vía Secrets Store CSI Driver, rotación y no inyección en variables cuando sea posible.
- Justificación: reduce riesgos de escape/privilegios y asegura cadena de suministro de contenedores.

17.7 DevSecOps y cadena de suministro

- Se usará CI/CD con escaneo IaC (Checkov/cdk-nag), SAST y DAST integrados al pipeline.
- Se usará SBOM generado (Syft) y firma de artefactos (cosign) antes de publicar en ECR; verificación en admisión (Sigstore/cosign-verify).
- Se usará CodeBuild/CodePipeline (o GitHub Actions) con roles mínimos y firmas de release.
- Justificación: integridad del software y cumplimiento de trazabilidad de cambios.

17.8 Detección y respuesta a incidentes

- Se usará AWS Security Hub como panel unificado; GuardDuty para detección (EKS/S3/EC2/Lambda/Malware Protection); CloudTrail y Config para cambios.
- Se usará Amazon Detective para análisis de causas; SNS para orquestar playbooks (ticketing/on-call) y AWS SSM para contención (aislar instancias/pods).
- Se usará runbooks de respuesta con evidencia (hashes, traceId, export a S3 WORM) y comunicación al regulador si aplica.
- Justificación: tiempos de detección y respuesta reducidos con evidencia preservada.

17.9 Registro, auditoría y conservación

- Se usará CloudTrail (todas las cuentas/regiones) con envío a S3 de seguridad (Object Lock) y CloudTrail Lake para análisis.
- Se usará VPC Flow Logs, ALB/CloudFront logs, KMS key access logs, y Cognito logs con retención y export a S3.
- Se usará la arquitectura de auditoría del punto 14 para no repudio de eventos de negocio.
- Justificación: trazabilidad completa para fines regulatorios, forenses y de mejora continua.

17.10 Cumplimiento sectorial

- Se usará mapeo de controles a ISO/IEC 27001/27002 y políticas internas del banco.
- Se usará PCI DSS si se almacenan/procesan datos de tarjeta; segmentación de redes, tokenización y PCI scope acotado.
- Se usará alineamiento con lineamientos de UAFE (AML/CFT) para retención de evidencias y reportes.
- Justificación: facilita auditorías y reduce riesgos regulatorios.

17.11 Criterios de aceptación

- MFA obligatorio para todo acceso de colaboradores; zero-trust en redes internas.
- Cifrado 100% en tránsito y reposo verificado; CloudTrail y Security Hub sin hallazgos críticos abiertos.
- Auditoría capaz de reconstruir transacciones y accesos con no repudio (punto 14).
- Privacidad: evidencias de gestión de consentimientos, retención y borrado conforme a LOPDP; sin PII en logs.

18. Costos y optimización

Objetivo: controlar y optimizar el gasto desde el diseño, con métricas por transacción, presupuestos y alertas, manteniendo la experiencia y los SLO definidos.

18.1 Gobierno FinOps y visibilidad

- Se usará AWS Budgets por cuenta/ambiente/dominio con alertas a SNS → Slack/Email a los dueños de cada servicio.
    - Justificación: detección temprana de desvíos y responsabilidad por área.
- Se usará AWS Cost Explorer y el Cost and Usage Report (CUR) a S3 (cuenta de seguridad), consultado con Athena/QuickSight.
    - Justificación: análisis detallado y reportes recurrentes sin mover datos.
- Se usará Cost Anomaly Detection con monitores por tag cost-center y servicio.
    - Justificación: alerta automática ante picos anómalos.

18.2 Etiquetado y contabilidad

- Se usará un esquema de etiquetas obligatorio: project, domain, env, owner, cost-center, compliance, data-class.
    - Justificación: imputación fina de costos y cumplimiento (p. ej., distinguir datos sensibles).
- Se usará AWS Config/SCPs para impedir recursos sin etiquetas.
    - Justificación: evita “gasto huérfano”.

18.3 Desglose de costos por capa (qué se usará y cómo optimiza)

- Canales (Web/Móvil): CloudFront + S3 (estáticos) y WAF/Shield.
    - Optimización: caché agresiva de estáticos, compresión, invalidation selectiva, TTFB bajo.
- Cómputo: Lambda (POV) y EKS (producción) con Karpenter/CA para autoescalado; Savings Plans (Compute) para capacidad base.
    - Optimización: rightsizing de pods/Nodos, Spot en no-prod, HPA por p95 de latencia.
- Datos transaccionales: Aurora PostgreSQL con RDS Proxy y Read Replicas.
    - Optimización: índices correctos, batching, evitar N+1, Serverless v2 en POV, reservas (RIs) en prod para capacidad estable.
- No transaccional / control: DynamoDB (idempotencia/outbox/locks) y Redis (caché/rate-limit).
    - Optimización: TTL en tablas de idempotencia, autoscaling de WCU/RCU, diseño de particiones uniformes; en Redis, evitar claves hot y TTL cortos.

- Archivos/evidencias: S3 con Object Lock (WORM) y lifecycle a Glacier.
    - Optimización: Intelligent-Tiering en buckets no WORM, compresión/parquet para logs, fusión de archivos pequeños.
- Mensajería/eventos: SQS FIFO y EventBridge.
    - Optimización: batch size adecuado, DLQ para aislar fallos y evitar reintentos infinitos.
- Notificaciones: SES (email), SNS Mobile Push (APNs/FCM), Twilio (SMS/WhatsApp, facturación externa).
    - Optimización: plantillas reutilizables, limitar adjuntos, controlar quiet hours para reducir envíos innecesarios.
- Observabilidad: CloudWatch/X-Ray/AMP/Grafana, Synthetics y RUM.
    - Optimización: retención de logs en 180 días hot → export a S3; muestreo de trazas; evitar métricas de alta cardinalidad.
- Red: VPC Endpoints para S3/DynamoDB/Secrets/KMS; Direct Connect a Core/BCE.
    - Optimización: minimizar NAT y transferencia cross-AZ, usar CloudFront para bajar egress.
- Seguridad: GuardDuty, Security Hub, Inspector, CloudTrail org.
    - Optimización: ingesta necesaria y suficiente; consolidación de cuentas de seguridad.

18.4 Unidades de costo (KPI FinOps)

Mediremos costo por transacción/unidad de negocio. Fórmulas (plantilla):

KPI	Fórmula	Meta inicial
Costo por 1 000 logins	(CloudFront + WAF + ALB/API + BFF compute) / (# logins / 1 000)	Definir con negocio
Costo por consulta de saldo	(BFF + caché Redis + Aurora lecturas + observabilidad) / # consultas	Definir
Costo por transferencia propia	(BFF + servicios TX + Aurora write + SQS/EventBridge + auditoría) / # transacciones	Definir
Costo por interbancaria	(anterior + BCE integración) / # interbancarias	Definir
Costo por notificación	(SES/SNS/Twilio + orquestación + auditoría) / # mensajes	Definir

Las metas se fijan con volúmenes proyectados y se revisan trimestralmente.

18.5 Presupuestos POV vs Producción

- POV/MVP: límite mensual por ambiente (Dev/QA/POV) en AWS Budgets; apagado automático nocturno en no-prod; Aurora Serverless v2 y on-demand en DynamoDB; trazas/logs con retención corta.

    - Justificación: pago por uso y control de experimentación.
- Producción: Savings Plans (1–3 años) para cómputo base; RIs de Aurora para capacidad estable; Global Tables/Aurora Global dimensionados a la demanda real; observabilidad con retención diferenciada (hot→cold).
    - Justificación: reducir costo recurrente sin perder elasticidad.

18.6 Alertas y revisiones

- Budgets: alertas al 70/90/100% del presupuesto mensual.
- Anomalías: Cost Anomaly Detection > 15% day-over-day o umbral absoluto por dominio.
- Revisión FinOps mensual: top 10 recursos por costo, servicios con picos, análisis egress, tamaño de colas/reintentos y hot keys en Redis/DynamoDB.

18.7 Criterios de aceptación

- 100% de recursos con etiquetas obligatorias; 0 recursos sin tag en auditoría de Config.
- Dashboards de unit economics publicados (QuickSight/Grafana) con tendencias de 90 días.
- Desvíos > 10% sobre tendencia tratados con acciones documentadas.
- POV dentro del presupuesto y producción con Savings Plans/RIs aplicados a ≥ 80% de la carga base.

19. CI/CD y gobernanza

Objetivo: entregar cambios seguros y trazables desde el commit hasta producción usando servicios nativos de AWS y segregación por cuentas/ambientes.

19.1 Repos y ramas

- Se usará AWS CodeCommit por dominio (front, back, infra, datos).
- Estrategia: trunk-based con feature branches cortas y PR obligatorio (2 revisores mínimo).
- Justificación: control de cambios simple, auditoría completa (CloudTrail) y menor fricción para releases frecuentes.

19.2 Orquestación multi-cuenta

- Se usará AWS CodePipeline en cuenta de herramientas y despliegue cross-account a dev / qa / stage / prod mediante IAM roles y KMS para artefactos.
- Justificación: separación de funciones (SoD), llaves por ambiente y trazabilidad única del flujo.

19.3 Artefactos y dependencias

- Se usará S3 (artifact bucket) con SSE-KMS y bloqueo de acceso público.
- Contenedores: Amazon ECR (escaneo con Inspector).
- Dependencias: AWS CodeArtifact para NPM/Maven/PyPI.
- Firma/SBOM: SBOM con Syft y firma de imágenes con cosign en CodeBuild; validación en admisión de EKS.
- Justificación: integridad de la cadena de suministro y evidencia de procedencia.

19.4 Pipelines (por tipo de componente)

a) Frontend Web (SPA)

- Build: CodeBuild (lint, unit tests, build).
- Empaquetado: subida a S3 (hosting estático).
- Entrega: invalidación de CloudFront (solo paths cambiados).
- Pruebas automáticas: CloudWatch Synthetics (Lighthouse básico, login, saldos).
- Aprobación manual stage→prod (doble control).
- Justificación: latencia baja y rollbacks inmediatos (versionado S3).

b) App Móvil

- Build firmado: CodeBuild con llaves en Secrets Manager (signing).
- Artefacto: APK/AAB/IPA a S3/ECR (para distribución interna).
- Gating: pruebas de UI en emulador y canarios API.
- Publicación: paso manual hacia las stores.
- Justificación: custodia segura de llaves y evidencia del binario entregado.

c) Backend en EKS

- Build & test: CodeBuild (unit + contract/Pact + integración).
- Contenedor: push a ECR; cosign firma; Inspector escaneo.
- Deploy: Helm/Kustomize desde CodeBuild a EKS (IRSA).
- Estrategia de despliegue: rolling con maxUnavailable=0, maxSurge=1; canary por ruteo ponderado (ALB Ingress Controller + reglas) y verificación de CloudWatch Alarms/X-Ray.
- Rollback automático si alarmas rojas.
- Justificación: despliegues seguros y reversibles con verificación en vivo.

d) Lambdas / API Gateway

- Deploy: CodeDeploy con canary (10%→100%) y rollback por alarma.
- Justificación: control fino de riesgo y reversión automática.

e) Infraestructura (IaC)

- Se usará AWS CDK con cdk-nag y pipelines de CodePipeline por stack.
- Validaciones: synth → diff → aprobación → deploy por ambiente.
- Justificación: cambios infra versionados, guardrails y auditoría de cada drift.

19.5 Calidad y seguridad (quality gates en pipeline)

- SAST (CodeBuild job), DAST (ZAP container contra stage), IaC lint (cdk-nag).
- Escaneo de imágenes (Inspector) y SBOM obligatorio.
- Políticas OPA/Gatekeeper (no root, FS read-only, límites de CPU/Mem) validadas en pre-deploy.
- Pruebas de contrato (Pact), performance (k6) y resiliencia (tests de retry/circuit).
- Justificación: prevenir vulnerabilidades y regresiones antes de tocar producción.

19.6 Migraciones de base de datos

- Se usará Flyway/Liquibase ejecutado por CodeBuild en etapa pre-deploy del servicio.
- Patrón: cambios compatibles hacia atrás (expand → deploy → contract).
- Aurora Blue/Green para cambios críticos en schema/procedimientos.
- Justificación: cero-downtime y reversión segura.

19.7 Configuración y secretos

- Se usará AWS AppConfig para feature flags y safe rollouts (1%→5%→100%).
- Configuración sensible: SSM Parameter Store y Secrets Manager (rotación).
- Inyección: por IRSA y runtime fetch (no build-time).
- Justificación: desacopla despliegue de activación funcional, reduce riesgo.

19.8 Gobernanza y controles de cambio

- Manual Approval stage→prod con dos aprobadores (Negocio + Operaciones/TI).
- Evidencias: PR link, resultados de pruebas, SBOM, resumen de riesgos.
- Bitácora: CodePipeline Execution History + CloudTrail + adjuntos en ticket.
- AWS Chatbot: notificaciones a Slack/Teams (aprobaciones y fallos).
- Justificación: cumplimiento de SoD y auditoría de cada release.

19.9 Observabilidad post-deploy y rollback

- Smoke tests automáticos (canarios) tras cada promoción.

- Gates por CloudWatch Alarms (latencia p95, 5xx, backlog SQS, lag Aurora).
- Rollback: CodeDeploy (Lambda) / `helm rollback` (EKS) / restore Blue/Green (Aurora).
- Justificación: detección temprana y recuperación rápida (MTTR bajo).

19.10 Criterios de aceptación

- 100% de servicios con pipeline CodePipeline y build reproducible.
- firma de contenedores y SBOM publicados por release.
- Aprobación dual y evidencias de pruebas en cada promoción a prod.
- Rollback automático verificado en drills trimestrales.
- Trazabilidad end-to-end (commit → imagen → despliegue → traceId) disponible para auditoría.

20. Pruebas y calidad

Objetivo: asegurar que la Banca por Internet cumpla funcionalidad, rendimiento, seguridad, accesibilidad y conformidad regulatoria antes de promocionar a producción, con gates automáticos en CodePipeline y evidencia auditada.

20.1 Estrategia integral

- Se usará una pirámide de pruebas: unitarias → contrato → integración → E2E/canarios, más ejes transversales (seguridad, performance, resiliencia, accesibilidad).
- Ambientes efímeros por PR (namespace EKS) para ejecutar integración/E2E aisladas.
- Datos de prueba: sintéticos o enmascarados (LOPDP), generados por job en CodeBuild; prohibido PII real en no-prod.

20.2 Unitarias (back/front)

- Backend: pruebas unitarias con cobertura ≥ 80% líneas/branches; mocks de Core/KYC/BCE.
- Frontend: unitarias de componentes y hooks; validación de formateo de importes y manejo de errores.
- Gate: falla el pipeline si la cobertura baja del umbral.

20.3 Pruebas de contrato (APIs y eventos)

- Se usará Pact para consumidor/proveedor en APIs REST (OpenAPI) y esquemas de EventBridge.
- Gate: contrato verificado en CI; no se permite breaking change.

20.4 Integración

- Se usará entorno efímero con docker-compose/Helm (servicios del dominio) y stubs de Core/BCE/KYC.
- Casos: idempotencia, sagas (compensaciones), retries/backoff, circuit breaker, consistencia entre Aurora y DynamoDB.

20.5 E2E funcional

- Se usará Playwright (web) y AWS Device Farm (móvil) para flujos: login/MFA, consulta saldos, transferencia propia e interbancaria, notificación.
- Gate: suite crítica 100% verde; evidencias (screenshots, HAR, videos) a S3.

20.6 Performance y capacidad

- Se usará k6 ejecutado en CodeBuild (o contenedores en EKS) con escenarios soak, stress y picos.
- Métricas objetivo (p95): login ≤ 800 ms, saldos ≤ 500 ms, propia ≤ 1.2 s, interbancaria ≤ 2.5 s.
- Telemetría: OTel/X-Ray, CloudWatch/Grafana; resultados y trend a S3/QuickSight.
- Gate: si p95 excede +20% del objetivo de forma sostenida, bloqueo de promoción.

20.7 Resiliencia y chaos

- Se usará AWS Fault Injection Service (FIS) para: caída de AZ, latencia/errores a Core/BCE/KYC, pérdida de red, saturación de colas, kill de pods críticos.
- Validaciones: circuit breaker abre/cierra, colas no pierden mensajes, compensaciones ejecutan, RTO interno dentro de objetivos.
- Gate: chaos experiments básicos deben pasar en stage antes de prod.

20.8 Seguridad (DevSecOps)

- SAST (CodeBuild job) y DAST (ZAP) contra stage;
- Container/IaC scanning: Amazon Inspector (ECR/Lambda/EC2) y cdk-nag/Checkov.
- Dependencias: verificación de CVEs y SBOM obligatorio; firma con cosign.
- OWASP ASVS nivel 2 como lista de control; pruebas de CSRF, XSS, SSRF, IDOR y autorización por scope/rol.
- Gate: cero hallazgos críticos/altos abiertos.

20.9 Accesibilidad y UX

- Se usará axe-core y Playwright para validar WCAG 2.1 AA (contraste, navegación teclado, aria).
- RUM: CloudWatch RUM con LCP/CLS/INP monitoreados.
- Gate: sin violaciones críticas y LCP p75 ≤ 2.5 s.

20.10 UAT y regulatorias (Ecuador)

- UAT con usuarios de negocio y mesa de riesgo/fraude. Casos: onboarding biométrico (FAR/FRR), límites, alertas antifraude, conciliación contable.
- Cumplimiento: evidencias LOPDP (consentimientos, retención/borrado), reportes para SB/UAFE cuando aplique.
- Notificaciones: pruebas con SES sandbox y Twilio (SMS/WhatsApp) en cuentas de prueba; revisión legal de textos/plantillas.

20.11 Observabilidad de pruebas

- Correlación con traceId/txnId; logs estructurados; captura de eventos de auditoría (punto 14) en escenarios de prueba.
- Centralización de reportes en S3 (Object Lock para evidencias clave) y panel Grafana/QuickSight por release.

20.12 Criterios de salida (Go/No-Go)

- Gates de seguridad, contrato, performance y E2E verdes.
- Defect leakage a stage < 2% en dos sprints.
- MTTR simulado ≤ 15 min (incidente P1) y plan de rollback validado.
- Aprobación dual (Negocio + Operaciones/TI) en CodePipeline con evidencia adjunta.

21. Roadmap de implementación

21.1 Fases y hitos (visión general)

Fase	Objetivo	Duración estimada	Entregables clave	Criterio de salida (Go/No-Go)
0. Preparación	Alinear alcance, riesgos, regulador y equipos	1–2 semanas	Plan del proyecto, matriz de riesgos, responsables, cronograma base	Aprobación de Negocio, Riesgos, TI y Seguridad
1. Descubrimiento & Diseño	Historias de usuario y experiencia;	2–3 semanas	Historias priorizadas, wireframes, C4	Revisión de stakeholders y señal verde de Seguridad

Fase	Objetivo	Duración estimada	Entregables clave	Criterio de salida (Go/No-Go)
	diagrama C4 y flujos principales		actualizado, plan de datos y seguridad	
2. POV (prueba de valor)	Probar la idea punta a punta a bajo costo	4–6 semanas	Login + consulta de saldos + transferencia propia en ambiente de prueba	Demostración funcional, métricas p95 aceptables, sin bloqueos críticos
3. MVP	Versión mínima utilizable por clientes	8–12 semanas	Web/App, saldos/movimientos, transferencias propias, notificaciones email/SMS, auditoría	UAT aprobado, pentest sin hallazgos críticos, mesa de ayuda lista
4. Piloto interno	Probar con colaboradores del banco	2 semanas	100–300 usuarios internos, manuales rápidos, canal de soporte	Tasa de éxito ≥ 95%, defectos corregidos, satisfacción ≥ 4/5
5. Piloto controlado (clientes)	Salir con un grupo pequeño real	4–6 semanas	1–5% de clientes, monitoreo 24/7, comunicación clara	SLO cumplidos (disponibilidad/latencia), quejas < 1%, sin incidentes de seguridad
6. Producción Fase 1	Go-Live amplio y estabilización	2 + 4 semanas	Lanzamiento general, operación 24/7, runbooks y on-call	Estabilización (4 semanas) con SLO ≥ 99.9%, sin reprocesos monetarios
7. Ampliaciones	Más funciones y canales	8–12+ semanas	Interbancarias, pagos de servicios, WhatsApp, mejoras UX, accesibilidad	Entregables por sprint con KPIs de uso y costo por transacción
8. Operación & Mejora continua	Optimizar y crecer	Permanente	Reportes mensuales, FinOps, seguridad continua, roadmap trimestral	Revisiones trimestrales con negocio y regulatorio interno

Notas: Duraciones son estimadas; se afinan con el banco. Cada fase incluye: seguridad, privacidad (LOPDP), accesibilidad y evidencia para auditoría.

21.2 Qué haremos en cada fase

- Preparación: alinear objetivos, armar el equipo, definir presupuesto y riesgos principales.
- Descubrimiento & Diseño: acordar la experiencia del cliente, dibujar “cómo conversa todo” (diagramas C4), y elegir qué entra primero.
- POV: montar lo mínimo en AWS para probar la idea end-to-end con pocos usuarios.
- MVP: construir lo necesario para que el cliente ya pueda usar banca en línea sin llamadas al banco.
- Pilotos: primero colaboradores, luego un grupo pequeño de clientes; medir, aprender y corregir.
- Producción Fase 1: salir para todos con monitoreo 24/7 y equipo de soporte listo.
- Ampliaciones: agregar pagos, interbancarias, WhatsApp, mejoras de usabilidad y accesibilidad.
- Operación: revisar métricas de negocio, costos y seguridad cada mes; planificar el siguiente trimestre.

21.3 Roles y responsabilidades

- Negocio: prioriza funcionalidades y aprueba mensajes/comunicaciones.
- Riesgos/Fraude: define umbrales, reglas y excepciones.
- Seguridad/Privacidad: revisa controles (MFA, cifrado, LOPDP).
- TI/Arquitectura: diseña y valida la solución en AWS.
- Desarrollo (Front/Back): construye y prueba.
- Operaciones/Soporte: runbooks, monitoreo y mesa de ayuda.
- Comunicaciones/Marketing: plan de lanzamiento y tutoriales.
- Legal/Compliance: valida textos, disclaimers y contratos.

21.4 Comunicación y adopción

- Antes del piloto: FAQs, tutorial en video (2–3 min), guía paso a paso.
- Durante piloto: canal abierto (WhatsApp/Chat del banco) y tiempos de respuesta claros.
- Go-Live: correos y notificaciones en App con mensajes simples (“qué hay de nuevo” y “cómo hacerlo”).
- Capacitación interna: sesiones de 1–2 horas para canales, call center y sucursales.

21.5 Puertas de calidad (Go/No-Go) en cada salida

- Funcional: casos críticos ok (login, saldos, transferencias).
- Rendimiento: p95 dentro de objetivo.

- Seguridad: sin hallazgos críticos.
- Soporte: mesa de ayuda y runbooks listos.
- Comunicación: materiales publicados.

22. Apéndices

22.1 Glosario (resumen práctico)

- BFF (Backend for Frontend): capa que adapta APIs al canal (Web/Móvil).
- OIDC/OAuth2 + PKCE: estándar de identidad/autorización con protección de intercambio de código.
- MFA / Step-up: segundo factor; exigido según riesgo/importe/acción.
- CQRS: separar lecturas (rápidas) de comandos (transaccionales).
- Idempotencia: misma orden → mismo resultado; evita dobles cargos.
- Outbox transaccional: cola persistida para publicar eventos sin perderlos.
- Saga/Compensación: orquesta pasos de una transacción; revierte si algo falla.
- RTO/RPO: tiempo objetivo de recuperación / pérdida máxima de datos.
- WORM (Object Lock): almacenamiento inmutable para evidencias.
- LOPDP: Ley Orgánica de Protección de Datos Personales (Ecuador).
- MRK: llaves multi-región de KMS para DR.
- SLO/SLI/SLA: objetivo/indicador/acuerdo de servicio.
- RACI: matriz de responsabilidades (Responsible, Accountable, Consulted, Informed).

22.2 Referencias normativas y técnicas (consulta)

- LOPDP (Ecuador) y guías de la Superintendencia de Bancos y UAFE (AML/CFT).
- AWS Well-Architected Framework (Security, Reliability, Cost, Operational Excellence, Performance).
- AWS Security Reference Architecture, Financial Services Lens.
- OWASP ASVS, NIST SP 800-63 (identidad).
- PCI DSS (si se tratan datos de tarjeta).

22.3 Mapeo a criterios de calificación (rastro rápido)

Criterio del evaluador	¿Dónde se cumple?
Requerimientos y justificación	Secc. 2–5, 7, 9–18
Diagramas C4 (calidad)	Secc. 6–8 + Anexo 22.2
Desacoplamiento/patrones	Secc. 7, 8, 11, 13–16
Front-end y móvil	Secc. 7–9, 12, 19–20
Datos y auditoría	Secc. 13–14
Conocimiento de AWS	Secc. 5, 7, 9–18

Criterio del evaluador	¿Dónde se cumple?
Costos/FinOps	Secc. 18
Autenticación	Secc. 9
Onboarding biométrico	Secc. 10
HA/FT/DR	Secc. 15
Monitoreo/Observabilidad	Secc. 16