Documento de Arquitectura del reto
DEVSU

Ricardo Rivera Guerra

Contenido

1. Introducción

Objetivo

El presente documento describe y justifica, en términos técnicos, la propuesta de arquitectura para el nuevo Sistema de Banca de BP. Su objetivo es permitir a los usuarios consultar sus movimientos y transferencias bancarias —internas y externas—, revisar su información y productos disponibles, mediante una aplicación web y otra móvil, garantizando a la vez el cumplimiento de las regulaciones y estándares de seguridad vigentes.

Alcance

El sistema cubrirá la consulta de saldos, movimientos y estados de cuenta; la ejecución de transferencias internas (entre cuentas propias y hacia terceros en BP) y externas a través de la red interbancaria; la visualización de productos disponibles, el envío de notificaciones transaccionales en tiempo real por correo electrónico, SMS y notificaciones "push"; el onboarding digital con verificación biométrica y registro de usuario; los procesos de autenticación, recuperación de credenciales y cambio de PIN o contraseña; así como la generación de una bitácora de auditoría y reportes regulatorios básicos.

2. Requisitos del sistema

Funcionales:

- Historial de transferencias bancarias.
- Información básica del cliente.
- Notificaciones.
- Onboarding biométrico de nuevos clientes
- Autenticación de acceso
- Auditoria y monitoreo de servicios

No funcionales:

- Alta disponibilidad y tolerancia a fallos.
- Escalabilidad y rendimiento.
- Seguridad y cumplimiento normativo (GDPR, protección de datos).

Tecnológicos:

Frontend:

- React 18 + Vite, ecosistema maduro, gran cantidad de librerías bancarias y compatibilidad total con PWAs.
- Flutter 3 (Dart), un solo código fuente nativo para Android e iOS; excelente soporte de widgets y rendimiento AOT.

Backend y Microservicios:

- Application Load Balancer, para gestionar el tráfico de solicitudes y balanceo de cargas entre nodos.
- Amazon EKS (Kubernetes gestionado), despliegue homogéneo, auto-escalado y zero-downtime releases.
- NestJS (TypeScript) para los microservicios, posee tipado fuerte, integración nativa con OpenAPI y fácil testing.
- Apache Kafka para el almacenamiento y ejecución de eventos

- Amazon SNS + SES para él envió de correos
- Firebase Cloud Messaging para el envio de notificaciones Push

Bases de Datos y Almacenamiento:

- Amazon RDS para datos relacionales para cuentas y transacciones
- MongoDB Atlas para coleccion registros de auditoría.
- Amazon S3 para almacenar documentos de identidad.
- ElastiCache Redis para almacenar data de consulta frecuente y almacenamiento de sesiones.

Integración Externa:

- Red Interbancaria (API REST), para transacciones interbancarias y externas
- Grafana para manejo de y control de métricas, logs y trazabilidad.

Seguridad:

- Amazon Cognito para autenticación segura con soporte OAuth 2.0.
- Amazon Rekognition Face para el manejo de la biometría del onboarding
- Amazon CLoudFront + WAF para protección OWASP
- TLS 1.3 end-to-end & datos cifrados en tránsito y en reposo para proteger datos sensibles.

Este conjunto de tecnologías permite una arquitectura escalable, segura y optimizada para ofrecer una experiencia de usuario fluida en todas las plataformas.

3. Descripción general

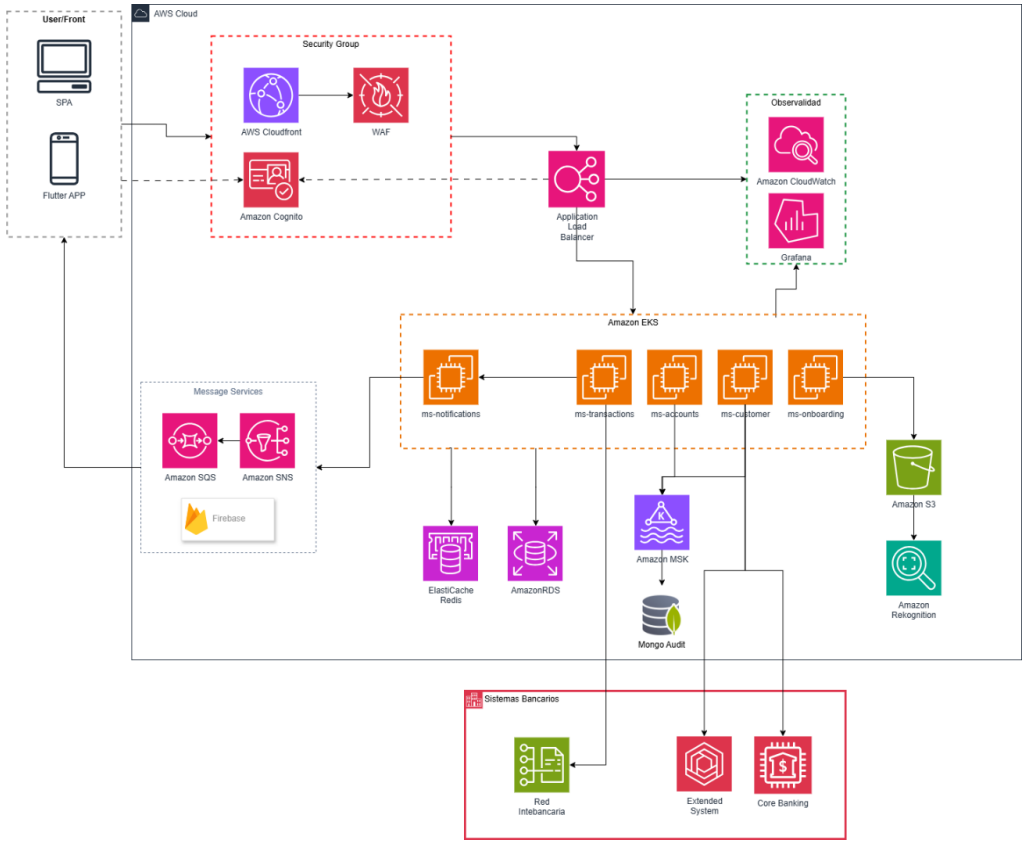


Diagrama general de arquitectura

Diagrama C4 (Contexto)

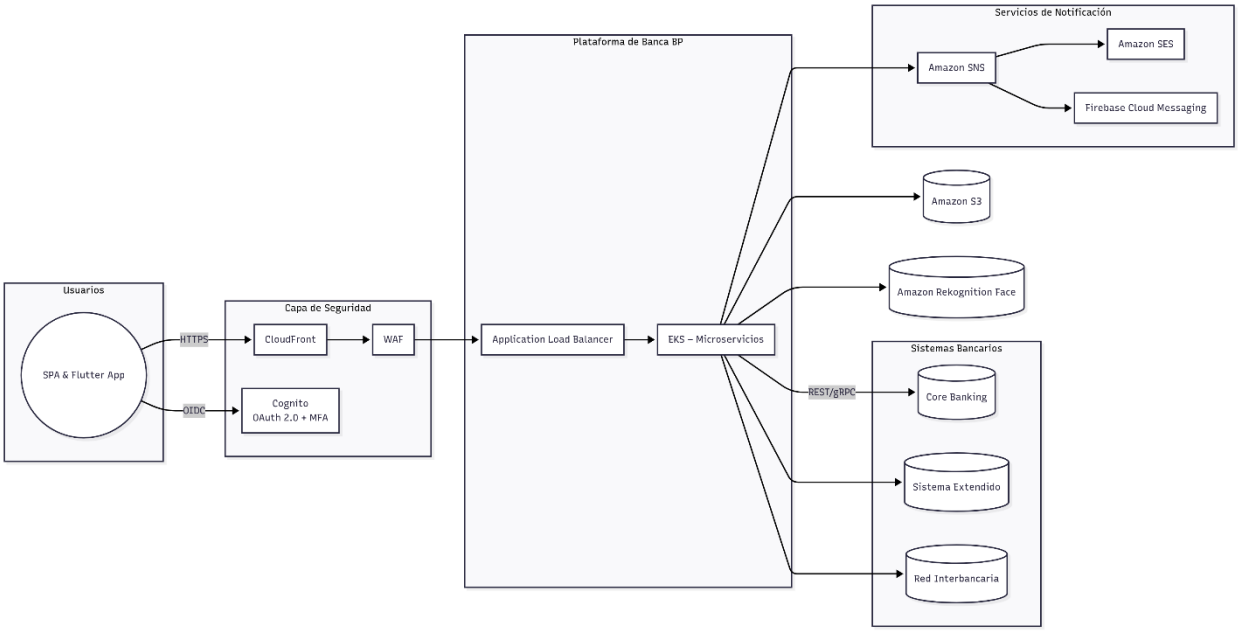


Diagrama C4 (Contexto o nivel 1)

Resumen de los componentes:

Se definieron a continuación los componentes resumidos del contexto general de la arquitectura:

- Aplicación web SAP (Single Page Application) y App Móvil
- Aplication Load Balancer y microservicios NestJS
- Bases datos (RDS y Mongo)
- Almacenamiento de documentos (S3) y servicios de verificación biométrica.
- Servicios de Seguridad (CloudFront y WAF)
- Servicios de envíos de Notificaciones y Correos
- Sistemas bancarios

4. Descripción de Arquitectura

Diagrama C4 (Contenedores)

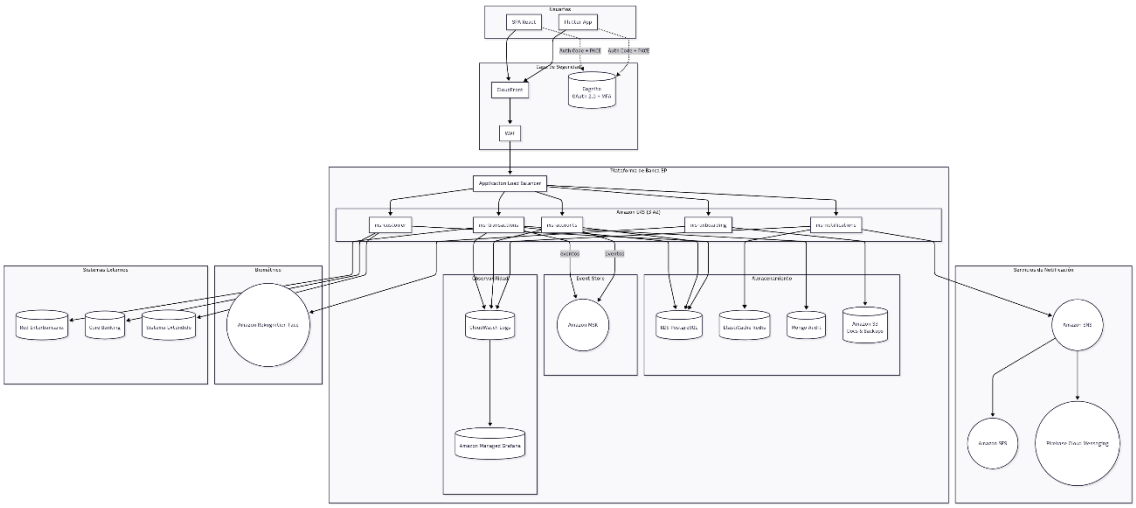


Diagrama C4 (Contenedores o nivel 2)

Descripción de Contenedores:

Frontend

- **Aplicación Web SPA (React 18 + Vite):** interfaz web donde el cliente consulta saldos, movimientos y realiza transferencias; PWA que se actualiza sin reinstalar y comparte tipado TypeScript con el back-end para acelerar el desarrollo.
- **Aplicación Móvil (Flutter 3):** app nativa multiplataforma Android/iOS que replica la funcionalidad web; el SDK de Flutter integra Firebase Cloud Messaging (FCM) para recibir notificaciones push en tiempo real.

Capa de seguridad

- **CloudFront + AWS WAF**: CDN perimetral con protección OWASP Top-10 y Shield Standard contra DDoS; reduce la latencia global sirviendo contenido estático cacheado.
- **Application Load Balancer (ALB)**: termina TLS 1.3, valida el JWT emitido por Cognito y distribuye tráfico L7 a los pods en EKS con sticky sessions opcionales.
- **Amazon Cognito**: proveedor OIDC/OAuth 2.0 que emite tokens (Auth Code + PKCE) y ofrece MFA; simplifica el ciclo de vida de usuarios y cumple PCI-DSS sin servidores propios.

Plataforma de Banca BP

- **ms-transactions**: orquesta transferencias internas / interbancarias y publica eventos de dominio en MSK; garantiza idempotencia y lleva control antifraude.
- **ms-accounts**: expone operaciones CRUD de cuentas y sincroniza saldos con el Core Banking; consume y emite eventos para mantener la consistencia eventual.
- **ms-customer**: consolida el perfil 360° del cliente consultando Core + Sistema Extendido, aplicando reglas KYC y límites de riesgo.
- **ms-onboarding**: gestiona el alta digital; captura documentos, invoca Amazon Rekognition para el match facial y guarda evidencias en S3.
- **ms-notifications**: genera mensajes transaccionales; publica en SNS para que los canales (SES / FCM) los distribuyan; usa Redis como rate-limiter.

Persistencia de Datos

- **Amazon RDS (PostgreSQL Multi-AZ)**: almacén ACID para cuentas, movimientos y clientes; fail-over automático garantiza RTO ≤ 2 h.
- **Amazon MSK (Serverless Kafka)**: event store para el patrón CQRS / Event Sourcing sin administrar brokers; topics particionados por agregado.
- **ElastiCache Redis**: caché de consultas calientes (< 20 ms P95) y contador de intentos para throttling de notificaciones.
- **MongoDB Atlas (colección *audit_log*)**: bitácora *append-only* con hash-chain firmada diariamente; coste casi cero y escalabilidad flexible.
- **Amazon S3**: almacena documentos de identidad y snapshots cifrados de RDS/Mongo en región secundaria, cumpliendo RPO ≤ 5 min.

Servicios de Notificación

- **Amazon SNS**: hub pub/sub que desacopla la lógica de negocio de los canales; fan-out garantiza al menos dos vías de entrega.
- **Amazon SES (e-mail)**: envía estados de cuenta y OTP firmados con DKIM/SPF, cumpliendo requisitos regulatorios.

- **Firebase Cloud Messaging (push)**: notificaciones móviles gratuitas y de baja latencia; se integra nativamente con Flutter.

Biométrico

- **Amazon Rekognition Face**: verifica identidad durante el onboarding; cumple ISO 30107-3 y evita mantener un motor de biometría propio.

Observabilidad

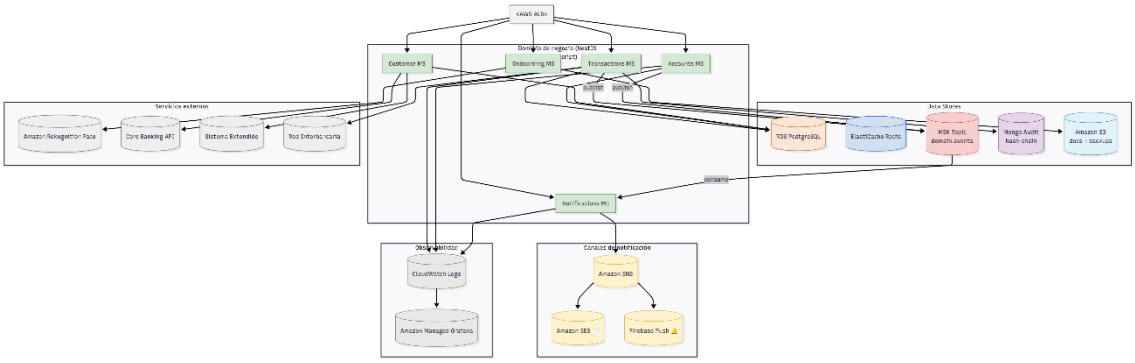
- **CloudWatch Logs**: centraliza registros y métricas, alimenta alarmas automáticas.
- **Grafana**: paneles de SLA/SLO unificados, permite a equipos de negocio y operaciones monitorear en tiempo real.

Sistemas Externos

- **Core Banking API:** fuente oficial de saldos y reglas contables; accedido por ms-customer y ms-transactions.
- **Sistema Extendido:** expone datos enriquecidos, consumido por ms-customer para el perfil 360°.
- **Red Interbancaria:** Gateway REST que procesa transferencias externas; integrado en ms-transactions con control de idempotencia y reintentos seguros.

5. Componentes detallados

Diagrama C4 (Componentes)



Microservicio: ms-customer (Consulta de Datos del Cliente)

- **Descripción:** Recupera el perfil 360 ° del cliente. Para datos esenciales consulta el Core Banking; para información enriquecida se conecta al Sistema Extendido.
- **Flujo de Datos:**
  1. El usuario solicita su perfil en la SPA o la app móvil.
  2. La petición viaja por CloudFront ➜ WAF ➜ ALB y llega al micro, portando un JWT de Cognito.
  3. ms-customer valida el token, consulta el Core Banking para datos básicos y, de ser necesario, el Sistema Extendido para completar la ficha.
  4. Devuelve el perfil consolidado al front y publica un evento CustomerProfileViewed en MSK para auditoría.

Microservicio: ms-accounts (Consulta de Estado de Cuentas)

- **Descripción**: Expone saldos, estados de cuenta y movimientos rápidos. Escucha eventos de otros micros para mantener los saldos al día.
- **Flujo de Datos**:

1. El usuario solicita el estado de sus cuentas.
2. La petición pasa por ALB y llega a ms-accounts.
3. El micro lee el saldo en RDS PostgreSQL; si el movimiento es reciente verifica el caché en Redis para responder en < 20 ms P95.
4. Si detecta actualizaciones pendientes, consume eventos en MSK (p. ej., TransactionCompleted) y actualiza los saldos antes de responder.

Microservicio: ms-transactions (Gestión de Transferencias)

- **Descripción**: Procesa transferencias internas (cuenta-propia y terceros BP) y externas vía Red Interbancaria. Garantiza idempotencia y registra la operación en la bitácora de auditoría.
- **Flujo de Datos**:

1. El cliente inicia una transferencia en la app.
2. La solicitud llega a ms-transactions a través del ALB; el micro valida fondos con ms-accounts.
3. Para transferencias internas: actualiza saldos en RDS dentro de una transacción ACID.
4. Para transferencias externas: invoca el API REST de la Red Interbancaria y maneja reintentos seguros.
5. Registra la transacción en RDS, genera evento TransactionCompleted en MSK y añade una entrada append-only en Mongo Audit.

Microservicio: ms-notifications (Gestión de Notificaciones)

- **Descripción**: Centraliza el envío de mensajes transaccionales y operativos, respetando cuotas y preferencias del usuario.
- **Flujo de Datos**:

1. Un evento relevante (p. ej., TransactionCompleted) llega desde MSK y despierta ms-notifications.
2. El micro verifica en Redis si el usuario superó su límite de notificaciones (rate-limit).
3. Publica el mensaje en Amazon SNS; el tópico fan-out envía:
    a. Correo vía Amazon SES (estados de cuenta, OTP).
    b. Push móvil vía Firebase Cloud Messaging.
4. Confirma el envío y escribe la evidencia en Mongo Audit.

Microservicio: ms-onboarding (Onboarding de Nuevos Clientes)

- **Descripción**: Facilita el proceso de incorporación de nuevos usuarios, validando su identidad con documentos y biometría. Los datos se almacenan en **Amazon S3** y se verifica con **Amazon Rekognition**. En casos donde se necesite asistencia de soporte durante el onboarding
- **Flujo de Datos**:

1. El usuario sube sus documentos y selfie desde la app.
2. Los archivos se guardan cifrados en Amazon S3.
3. El micro invoca Amazon Rekognition Face para validar la coincidencia rostro-documento.
4. Si la verificación es exitosa, crea el usuario en Cognito, registra el alta en RDS y envía un evento CustomerOnboarded a MSK.

5. Almacena un hash del proceso en Mongo Audit para no repudio.

6. Cumplimiento y Seguridad

Cumplimiento Normativo:

- **Protección de Datos:** Cumple la Ley Orgánica de Protección de Datos Personales (EC) y los principios de GDPR; todos los datos personales se tratan con consentimiento explícito y propósito limitado.
- **Registros de Auditoría:** Cada acción relevante se registra en la colección *audit_log* de MongoDB Atlas como append-only y archivado en S3 Glacier para conservar evidencia ≥ 10 años.
- **Retención de Datos:** Políticas automáticas en S3 Lifecycle y RDS para borrar o anonimizar datos una vez cumplido el plazo legal; avisos al usuario cuando se modifique su información o preferencia.
- **Pagos y Tarjetas:** Encriptación de PAN y segregación de redes cumplen los controles centrales de **PCI-DSS 4.0** (secciones 3, 4, 7 y 10).

Seguridad del Sistema:

- **Autenticación Segura:** Amazon Cognito con flujo OAuth 2.0 (Auth-Code + PKCE) y MFA (TOTP o push). El onboarding usa verificación biométrica con Amazon Rekognition.
- **Cifrado Extremo a Extremo:** TLS 1.3 para datos en tránsito; AES-256 en reposo en RDS, MongoDB Atlas, MSK, Redis y S3 (claves gestionadas por KMS).
- **Gestión de Credenciales:** AWS Secrets Manager mantiene y rota secretos de bases de datos, Redis y FCM; acceso mediante IAM Roles for Service Accounts (IRSA).
- **Control de Acceso (RBAC):** Grupos Cognito y políticas IAM “least-privilege” limitan lo que cada rol de microservicio o usuario puede ver o hacer.
- **Monitoreo y Alertas:** Logs y métricas centralizadas en CloudWatch; paneles en Grafana; alertas a PagerDuty cuando latencia > 300 ms P95 o errores > 1 %.
- **Protección DDoS y OWASP:** AWS Shield Standard + WAF con reglas OWASP Top-10 y limitación de 5 k RPM por IP.
- **Hardening de Contenedores:** Imágenes distroless escaneadas en ECR; políticas PodSecurity, seccomp y AppArmor activas; fallos críticos bloquean el despliegue CI/CD.

7. Escalabilidad y Alta Disponibilidad

Escalabilidad:

- **EKS auto-scalable**: HPA aumenta réplicas de pods y Karpenter agrega nodos EC2 cuando la carga lo exige.
- **Edge elástico**: CloudFront + ALB se expanden automáticamente y absorben picos sin intervención.
- **MSK Serverless & Redis clúster**: eventos y caché crecen horizontalmente bajo demanda.
- **Bases de datos**: RDS se expande verticalmente o con read-replicas; Mongo Atlas permite sharding cuando aumenta la bitácora.

Alta Disponibilidad:

- **Multi-AZ en todo**: EKS, ALB, RDS Multi-AZ y Mongo replica-set distribuidos en ≥ 2 zonas.
- **Backups y DR**: Snapshots automáticos hora-a-hora de RDS y export diarios de Mongo se copian vía S3 Cross-Region Replication, cumpliendo RPO ≤ 5 min y RTO ≤ 2 h.
- **Auto-healing**: Kubernetes reinicia pods fallidos; los *loaders* de EKS sustituyen nodos no saludables sin intervención humana.
- **Protección perimetral**: Shield Standard + WAF mitigan DDoS y ataques OWASP, preservando servicio continuo y límites de solicitud (WAF rate-based rules) y *circuit-breakers* en ms-notifications evitan sobrecargar servicios downstream.

8. Costos estimados

Componente	Configuración	USD/mes aprox.
EKS	2 × t3.medium × 3 AZ + control plane	$ 430,00
RDS PostgreSQL Multi-AZ	db.m6g.large	$ 380,00
MSK Serverless	1 GB/s ingest	$ 200,00
Mongo Audit	Atlas M2 replica set	$ 15,00
ElastiCache Redis	cache.t3.micro	$ 20,00
SNS + SES	100 k correos	$ 10,00
FCM Push	—	$ -
CloudFront + WAF	1 TB egress + 10 M req.	$ 85,00
Grafana	—	$ -
Backups S3	250 GB IA + Glacier	$ 12,00

La plataforma, desplegada íntegramente en AWS y complementada con servicios gratuitos de Google (FCM) y Mongo Atlas, implica un gasto operativo aproximado de **USD 1 150 al mes**.
Este total resulta de los principales rubros productivos:

- EKS multi-AZ (≈ 430 $)
- RDS PostgreSQL Multi-AZ (≈ 380 $)
- MSK Serverless (≈ 200 $)
- Mongo Audit, Redis, CloudFront + WAF, respaldos S3 y correos SES/SNS (≈ 140 $ en conjunto)