Arquitectura de Solución: Sistema de Banca por Internet para BP

1. Resumen Ejecutivo

Este documento presenta la arquitectura propuesta para el sistema de banca por internet de la entidad financiera BP. El objetivo es permitir a los usuarios consultar su historial de movimientos, realizar transferencias y efectuar pagos entre cuentas propias e interbancarias, utilizando una plataforma segura, moderna y de alta disponibilidad. La solución considera requerimientos normativos de la industria financiera, incorpora múltiples capas de seguridad, y está diseñada con un enfoque escalable y desacoplado que permite una fácil extensión y mantenimiento a futuro.

2. Justificación de decisiones arquitectónicas

Frontend Web (SPA): Angular

- *Justificación 1:* Angular es un framework mantenido por Google, con una arquitectura modular que facilita el desarrollo escalable, seguro y estructurado. Su madurez lo convierte en una opción sólida para proyectos de nivel empresarial.
- *Justificación 2:* Integra de forma eficiente con APIs RESTful, bibliotecas de seguridad, y proveedores de autenticación OAuth2, lo cual permite una implementación rápida y confiable del frontend bancario.

Aplicación Móvil: React Native

- *Justificación 1:* Permite el desarrollo de apps nativas para iOS y Android con una sola base de código, reduciendo tiempos y costos de desarrollo.
- *Justificación 2:* Cuenta con una amplia comunidad, integración nativa de SDKs y herramientas para incorporar funcionalidades críticas como el reconocimiento facial para onboarding.

Backend: Arquitectura de Microservicios sobre PHP (Laravel o Symfony)

- *Justificación 1:* PHP con Laravel o Symfony permite estructurar microservicios de manera eficiente, utilizando controladores REST, colas, eventos y acceso a bases de datos relacionales y no relacionales.

- *Justificación 2:* Su ecosistema maduro y documentado, junto con herramientas de testing y monitoreo, favorece la entrega continua y el despliegue automatizado de servicios.

OAuth2.0 + Identity Provider (ej. Keycloak / Auth0)

- *Justificación 1:* OAuth2 es el estándar más adoptado para control de acceso y autorización en sistemas distribuidos.
- *Justificación 2:* Usar un Identity Provider externo como Auth0 o Keycloak permite delegar la complejidad de la gestión de sesiones, tokens, MFA y flujos de onboarding seguro.

Notificaciones: Amazon SNS + Firebase Cloud Messaging

- *Justificación 1:* Estas soluciones permiten gestionar notificaciones multicanal (push, email, SMS) con alta disponibilidad y escalabilidad global.
- *Justificación 2:* Son servicios gestionados que permiten configurar reglas de entrega, reintentos y auditoría de mensajes.

Base de datos de auditoría: PostgreSQL o Elasticsearch

- *Justificación 1:* PostgreSQL garantiza integridad transaccional y es ideal para queries estructuradas, mientras que Elasticsearch permite indexación eficiente para consultas complejas de logs.
- *Justificación 2:* Ambas se integran fácilmente con herramientas de visualización y monitoreo como Kibana o Grafana.

Infraestructura: AWS

- *Justificación 1:* AWS provee servicios gestionados para microservicios, balanceo de carga, bases de datos, seguridad y backups, lo cual permite enfocarse en el negocio y no en la infraestructura.
- *Justificación 2:* Soporte para múltiples zonas de disponibilidad, autoescalado, y cumplimiento de normativas como PCI-DSS, ISO 27001 y SOC 2.

3. Diagramas C4

Diagrama de Contexto (Figura 1)**:** Representa a los usuarios interactuando con la SPA web y la app móvil, los cuales se comunican con el sistema core bancario, sistemas externos, un gateway de API, el sistema de notificación y el servicio de auditoría. Es ideal para presentar la solución a perfiles no técnicos.

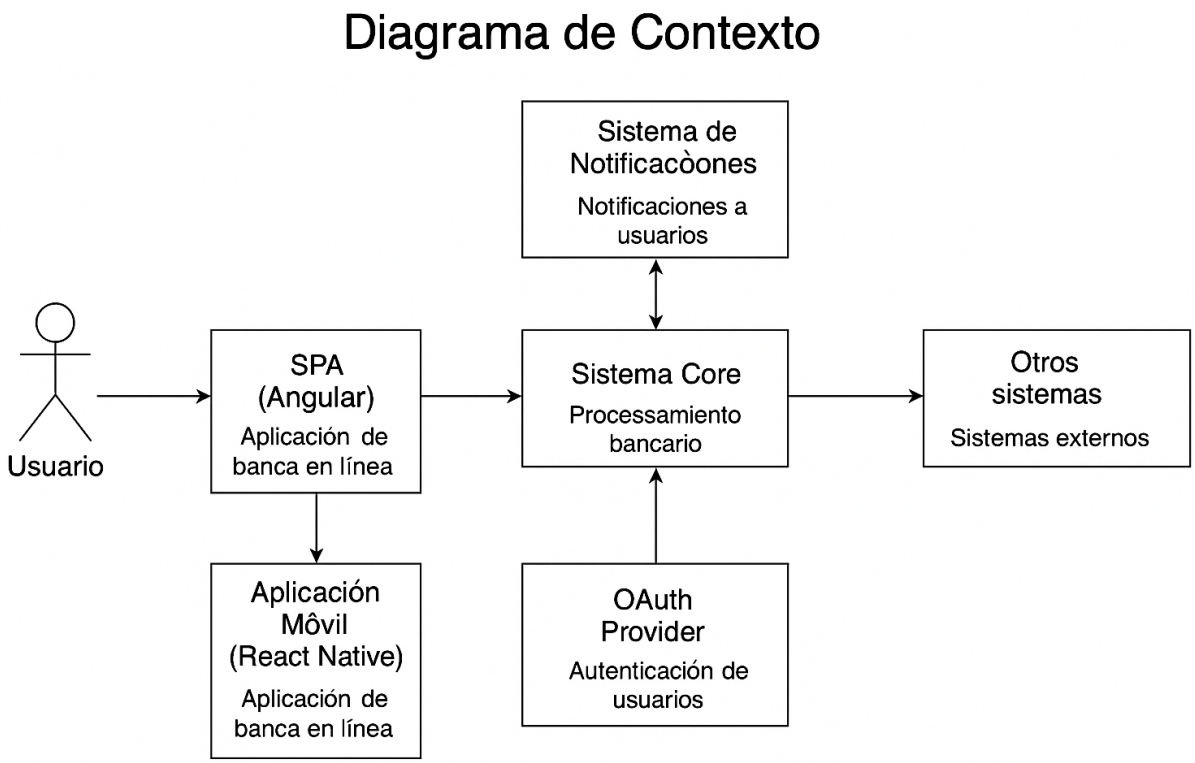


Diagrama de Contenedores (Figura 2): Muestra los principales contenedores lógicos: la SPA (Angular), la app móvil (React Native), el API Gateway, los microservicios principales (movimientos, transferencias, onboarding, autenticación, notificaciones y auditoría), el proveedor OAuth, las bases de datos y las integraciones externas.

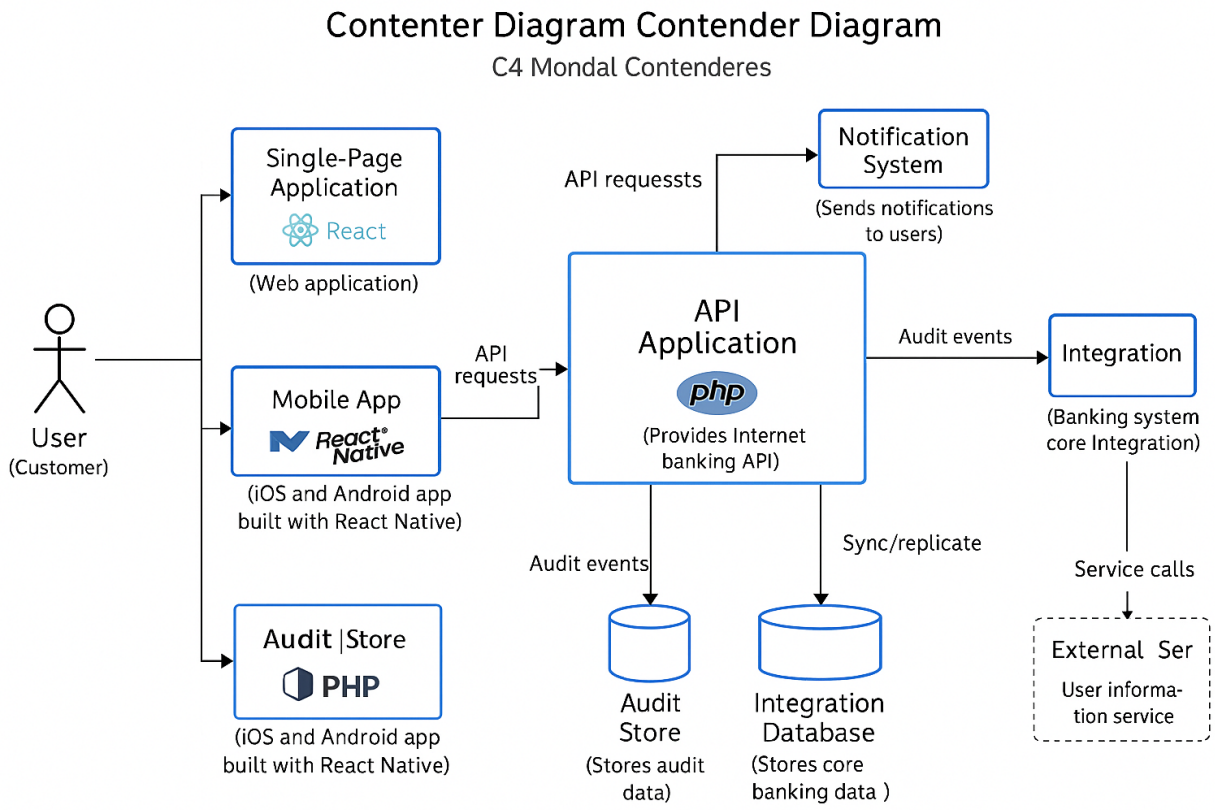
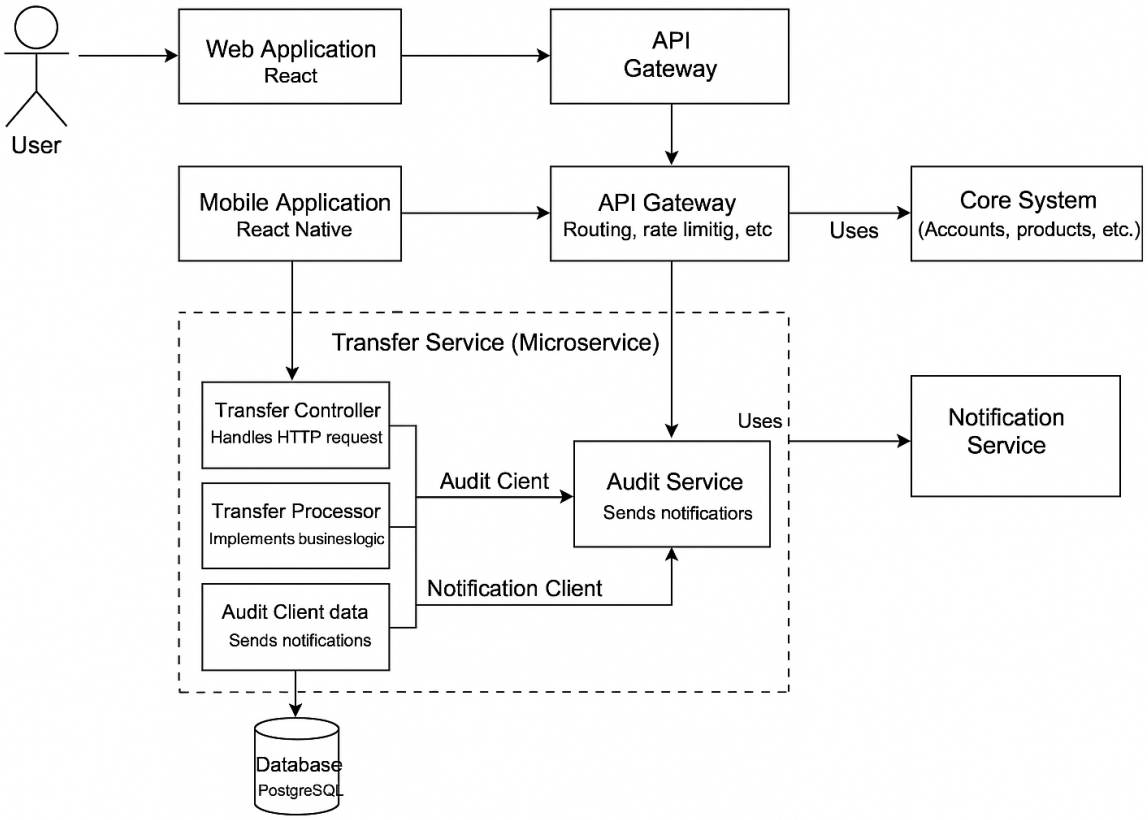


Diagrama de Componentes (Figura 3): Se detalla el microservicio de transferencias, incluyendo su controlador REST, el procesador de negocio, validaciones antifraude, clientes de integración con el core bancario, cliente de auditoría, cliente de notificaciones, y su base de datos PostgreSQL.



4. Autenticación y Onboarding

El sistema implementará un flujo OAuth2 utilizando Authorization Code Flow con PKCE para el acceso seguro desde SPA y móvil. El proceso de onboarding para nuevos usuarios móviles incluirá un módulo de reconocimiento facial usando AWS Rekognition o Azure Face API, comparando el rostro en tiempo real contra la imagen de la cédula o pasaporte.

El flujo recomendado incluye:

- Registro del usuario con captura de rostro y documentos.
- Validación de identidad contra servicios externos.
- Generación de credenciales seguras y entrega de token OAuth2.
- Activación del perfil y almacenamiento del registro de auditoría.

5. Auditoría y Persistencia

Todas las acciones realizadas por los usuarios y por los sistemas automáticos serán registradas en un servicio de auditoría que consolida la información en una base de datos optimizada. Esta solución garantizará trazabilidad y cumplimiento normativo mediante:

- Uso de eventos de auditoría por microservicio.
- Persistencia estructurada y segura de logs.
- Consultas rápidas para auditor interno mediante dashboards (Kibana o Grafana).

6. Consideraciones Normativas

La solución cumple con la Ley Orgánica de Protección de Datos Personales del Ecuador (LOPDP), incluyendo los principios de minimización de datos, seguridad por diseño y por defecto, y derechos del titular. Además, se consideran:

- Encriptación en tránsito (TLS 1.3) y en reposo (AES-256).
- Autenticación multifactor (MFA) y expira automática de sesiones.
- Cumplimiento con ISO 27001, PCI-DSS y OWASP Top 10.

7. Alta disponibilidad y monitoreo

La arquitectura está diseñada para garantizar continuidad operativa, con:

- Despliegue multi-AZ (alta disponibilidad) en ECS/Fargate.
- Balanceo de carga automático y autoescalado según demanda.
- Backups en S3 con retención cifrada y replicación entre regiones.
- Monitoreo con CloudWatch Logs, Prometheus y Grafana.
- Alarmas de salud, métricas de rendimiento y dashboard de disponibilidad.

8. Conclusión

La arquitectura propuesta combina buenas prácticas de ingeniería de software, seguridad de la información y cumplimiento normativo. Permite escalar según demanda, integrar nuevos módulos o servicios, y responder ante incidentes con resiliencia. La elección de componentes desacoplados, patrones de integración moderna, y servicios en la nube permite al banco BP mantenerse competitivo y preparado para innovar.

Repositorio GitHub: https://github.com/juanitourquiza/bp-online-banking-architecture

Autor: Juan Urquiza

Fecha: Octubre 2025