Arquitectura de Solución para Sistema de Banca por Internet - BP

1. Introducción

Esta propuesta arquitectónica tiene como objetivo resolver el desafío de diseñar un sistema moderno de banca por internet que cumpla con criterios de alta disponibilidad, seguridad, interoperabilidad y cumplimiento regulatorio. El sistema debe ofrecer una experiencia consistente a través de canales digitales (web y móvil), integrar sistemas legados, permitir la trazabilidad de operaciones mediante auditoría, y facilitar la escalabilidad y el mantenimiento continuo mediante microservicios y servicios cloud.

La entidad BP ha encomendado el diseño de una arquitectura de soluciones para un sistema de banca digital, el cual debe permitir a los usuarios acceder a su información financiera, consultar movimientos, realizar transferencias entre cuentas propias e interbancarias, recibir notificaciones y garantizar cumplimiento normativo.

Esta propuesta se desarrolla con base en el marco de arquitectura empresarial TOGAF - ADM y se describe técnicamente bajo el modelo de arquitectura C4. La solución contempla integraciones con sistemas legados, servicios cloud, arquitectura de microservicios y principios de seguridad, escalabilidad y alta disponibilidad.

2. Metodología de Desarrollo Arquitectónico

Esta propuesta está basada en el marco de ***Arquitectura Empresarial TOGAF - ADM (Architecture Development Method)***, asegurando alineación con los objetivos estratégicos de BP, interoperabilidad con sistemas existentes y optimización de costos operativos.

2.1 TOGAF ADM Aplicado

Se ha seguido el ciclo de vida ADM para guiar el diseño:

- **Fase A - Visión de Arquitectura:** Objetivos: seguridad, escalabilidad, cumplimiento, experiencia omnicanal.
- **Fase B - Negocio:** Identificación de actores, casos de uso (consultas, pagos, onboarding).
- **Fase C - Sistemas de Información:** Microservicios, SPA, app móvil, core bancario, notificaciones.
- **Fase D - Tecnología:** API Gateway, servicios cloud, base de auditoría, balanceo y DR.
- **Fases E/F - Plan de Transformación**: Roadmap incremental en entornos dev/QA/prod.
- **Fase G - Gobernanza:** KPIs, monitoreo, revisión de SLA y cumplimiento normativo.
- **Fase H - Cambios:** Modelo evolutivo con capacidad de crecimiento y adaptabilidad.

3. Justificación de Decisiones Arquitectónicas

Esta sección detalla las decisiones clave tomadas durante el diseño de la solución, junto con su justificación técnica, fundamentadas en principios de arquitectura empresarial, seguridad, rendimiento y escalabilidad.

3.1 Autenticación - OAuth 2.0 + PKCE

Esta elección responde al requerimiento de un flujo de autenticación seguro y estandarizado:

- **OAuth 2.0** permite delegar autorización sin exponer credenciales de usuario al cliente, separando control de acceso del backend.
- **PKCE (Proof Key for Code Exchange)** añade seguridad contra interceptación de tokens en entornos móviles o SPAs, reemplazando el uso de client secrets.
- Se alinea con recomendaciones OWASP y es ideal para clientes públicos sin backend confidencial.

3.2. Aplicaciones Front-End

La propuesta utiliza tecnologías modernas orientadas a experiencia de usuario, rendimiento y portabilidad:

- **SPA Web (React)** permite interfaces reactivas, modularidad, y una arquitectura desacoplada mediante consumo de APIs REST.
- **Aplicación Móvil (Flutter)** permite desarrollo multiplataforma (Android/iOS) con alta performance nativa, reduciendo tiempos y costos de desarrollo. Integra fácilmente funcionalidades como autenticación biométrica y notificaciones.

3.3. Onboarding Biométrico

Se propone incorporar un mecanismo moderno de validación de identidad:

- **Servicios como AWS Rekognition, Azure Face API o Face++** permiten verificar rostros contra documentos o plantillas previas.
- Se integra al flujo de onboarding para clientes nuevos o como factor de verificación para operaciones críticas.
- Mejora la experiencia del usuario al eliminar barreras de ingreso y fortalece los controles antifraude.

3.4. Integración y Microservicios

El diseño modular se fundamenta en principios de arquitectura desacoplada:

- **API Gateway (Kong, NGINX, AWS API Gateway)** actúa como punto único de entrada, gestiona autenticación, rate limiting y logging.
- **Cada microservicio** maneja un dominio funcional (por ejemplo, autenticación, auditoría, movimientos) y comunica vía HTTP/REST o eventos, facilitando despliegues independientes y escalabilidad.
- Se alinea con patrones como SRP (responsabilidad única) y 12-Factor App.

3.5. Consideraciones de Seguridad y Cumplimiento Normativo

Se contemplan dos componentes clave para trazabilidad y rendimiento:

- **Auditoría (PostgreSQL dedicada):** se registran eventos críticos del sistema (transacciones, accesos, errores) para cumplir con normativas como ISO 27001 y LOPD/GDPR. Los datos no se mezclan con transacciones operativas.

- **Cache Redis (Write-through):** para usuarios frecuentes o cuentas activas se cachean datos como saldos o movimientos, acelerando consultas y reduciendo carga al core bancario, sin sacrificar integridad.

4. Requisitos de Calidad y Consideraciones

- **Alta Disponibilidad:** Replicación en múltiples zonas de disponibilidad.
- **Tolerancia a Fallos / DR:** Despliegue en nubes con balanceadores, backups automáticos.
- **Seguridad:** TLS 1.3, MFA, cifrado en reposo y en tránsito.
- **Normativas:** LOPD/GDPR, ISO 27001, PCI-DSS, PSD2.
- **Escalabilidad:** Kubernetes/EKS o AKS, autoescalado, desacoplamiento horizontal.
- **Monitoreo:** Prometheus + Grafana, logs con ELK Stack o AWS CloudWatch.

5. Beneficios de la Implementación

Beneficios Estratégicos

a. **Mejora en la experiencia del cliente:** Acceso rápido y seguro a servicios bancarios digitales.
b. **Mayor competitividad:** Permite a BP mantenerse a la vanguardia en tecnología financiera.
c. **Interoperabilidad con sistemas legados:** Facilita la transición sin afectar operaciones actuales.
d. **Cumplimiento normativo:** Adherencia a regulaciones bancarias nacionales e internacionales.

Beneficios Técnicos

a. **Alta disponibilidad y tolerancia a fallos:** Arquitectura escalable y redundante.
b. **Automatización de procesos:** Integración con IA para optimización de operaciones.
c. **Seguridad reforzada:** Implementación de MFA, cifrado de datos y monitoreo proactivo.
d. **Eficiencia en costos:** Uso de infraestructura híbrida con optimización de recursos.

6. Riesgos y Mitigación

Riesgos Potenciales

1. **Resistencia al cambio:** Integración con sistemas legados puede generar fricción interna.
2. **Seguridad y fraude:** Posibles ataques cibernéticos y vulnerabilidades de autenticación.
3. **Costos de implementación:** Inversión inicial alta para infraestructura y capacitación.
4. **Interoperabilidad compleja:** Integración con múltiples sistemas puede generar retrasos.

Estrategias de Mitigación

a. **Plan de gestión del cambio:** Capacitación y adopción progresiva en la organización.
b. **Seguridad avanzada:** Implementación de Zero Trust Architecture y monitoreo de amenazas.
c. **Optimización financiera:** Estrategia de implementación por fases para control de costos.
d. **Pruebas continuas:** Uso de pruebas de estrés y validaciones en entornos sandbox.

7. Integración con Sistemas Legados

Estrategia de Interoperabilidad

Dado que BP es el mayor banco del país y posee múltiples sistemas heredados, la arquitectura se diseñará con una capa de integración basada en:

- **API Gateway con adaptación para servicios SOAP y REST:** Permitiendo la comunicación entre sistemas modernos y legados.
- **Enterprise Service Bus (ESB):** Para facilitar la orquestación de servicios y transformar formatos de datos entre sistemas antiguos y nuevos.
- **ETL y Middleware:** Para sincronización de datos y reducción de latencia en consultas.
- **Mensajería Asíncrona (Kafka o RabbitMQ):** Para la propagación eficiente de eventos y sincronización de información entre plataformas.

8. Aplicaciones de Inteligencia Artificial

La integración de inteligencia artificial permite optimizar operaciones y mejorar la experiencia del usuario. Algunas aplicaciones incluyen:

- **Detección de Fraudes:** Uso de modelos de machine learning para identificar patrones de transacciones sospechosas en tiempo real.
- **Chatbots y Asistentes Virtuales:** Implementación de IA conversacional con NLP para atención al cliente.
- **Análisis Predictivo:** Modelos para anticipar necesidades del usuario y ofrecer productos financieros personalizados.
- **Optimización del Onboarding:** Validación biométrica y detección de riesgos en tiempo real mediante IA.
- **Automatización de conciliaciones financieras:** Reducción de errores en la sincronización de datos entre sistemas nuevos y legados.

9. Fundamentación Técnica para Abordar las Consideraciones

Alta disponibilidad y tolerancia a fallos

- **Uso de balanceadores de carga:** AWS Elastic Load Balancer o Azure Load Balancer.
- **Replicación de bases de datos:** PostgreSQL con streaming replication o MySQL con Group Replication.

- **Orquestación de contenedores:** Kubernetes (EKS en AWS o AKS en Azure) con autoescalado.
- **Failover y Recuperación ante Desastres (DR):** Replicación geográfica de bases de datos y respaldos automáticos en múltiples regiones.

Seguridad

- **Autenticación multifactor (MFA):** Integración con herramientas como Auth0, Keycloak o AWS Cognito.
- **Cifrado de datos:** Uso de TLS 1.3 para datos en tránsito y AES-256 para datos en reposo.
- **Control de acceso basado en roles (RBAC):** Aplicado en los microservicios mediante Open Policy Agent (OPA).
- **Detección y respuesta a amenazas:** Implementación de WAF y SIEM (Splunk, AWS GuardDuty o Azure Sentinel).

Escalabilidad y sustentabilidad del negocio

- **Arquitectura híbrida:** Integración con infraestructura on-premise y cloud para garantizar flexibilidad y costos optimizados.
- **Uso de contenedores:** Kubernetes para despliegues ágiles y eficientes.
- **Microservicios desacoplados:** Permiten crecimiento modular del sistema.
- **Optimización de costos:** Uso de instancias serverless y gestión eficiente de bases de datos con almacenamiento elástico.

Sincronización y persistencia

- **Redis o DynamoDB:** Para mejorar tiempos de respuesta en clientes frecuentes.
- **Data Lake y Big Data:** Uso de AWS S3 o Azure Data Lake para almacenamiento a gran escala.
- **Integración con bases de datos legadas:** Uso de Oracle Exadata o DB2 para garantizar la continuidad operativa.
- **Batch Processing:** Para procesar grandes volúmenes de datos en horarios fuera de pico.

Observabilidad y Monitoreo

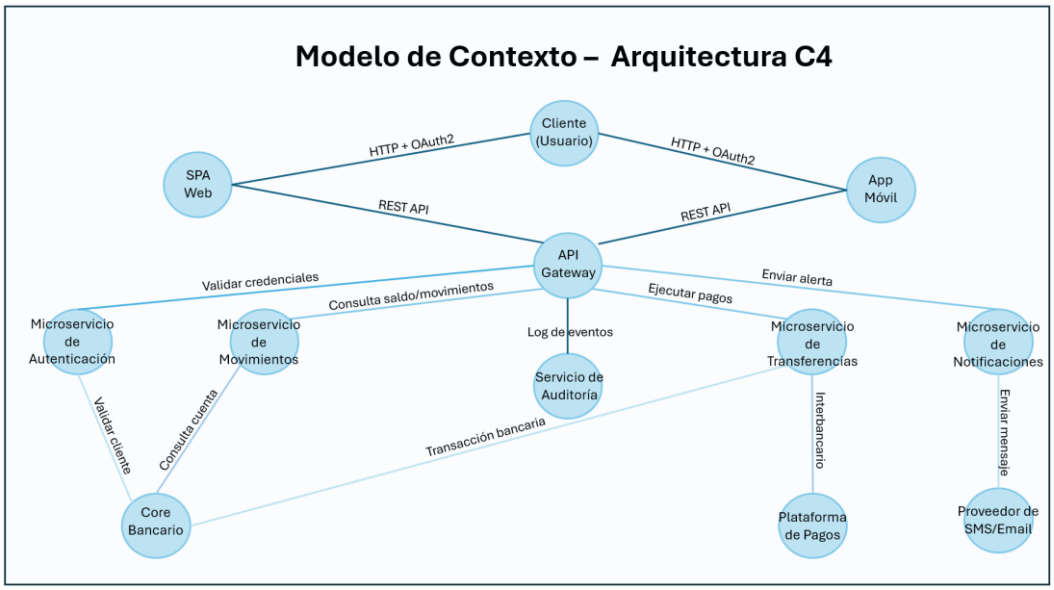
- **Prometheus y Grafana:** Para métricas en tiempo real.
- **AWS CloudWatch o Azure Monitor:** Para logs y alertas proactivas.
- **Distributed Tracing con OpenTelemetry:** Para seguimiento de transacciones en microservicios.
- **AI Ops para optimización de rendimiento:** Implementación de IA para detección proactiva de anomalías en la infraestructura.

10. MODELOS

a. **Modelo de Contexto:**

El modelo de contexto representa a alto nivel cómo los usuarios interactúan con los canales digitales del banco (SPA y aplicación móvil), y cómo estas plataformas se conectan con los microservicios a través del API Gateway. Los microservicios, a su vez, interactúan con sistemas externos como el Core Bancario, la Plataforma de Pagos y el proveedor de notificaciones. Las flechas

del diagrama están etiquetadas con tecnologías como REST, OAuth2, y HTTPS para indicar protocolos seguros de comunicación. Esta capa permite a los interesados no técnicos comprender el alcance global del sistema.



El **Modelo de Contexto** del sistema de banca digital representa una vista de alto nivel donde se identifican los actores clave y sus interacciones con los distintos sistemas involucrados. Este modelo permite entender **qué entidades** interactúan con la plataforma y **cómo lo hacen**, sin profundizar aún en los detalles de implementación. A continuación, se describe cada uno de los componentes y sus relaciones:

Componentes del Modelo de Contexto

a. **Cliente (Usuario Final)**
   - Es el actor principal del sistema. Representa a los usuarios que interactúan con la banca digital a través de distintos canales.
   - Puede ser un cliente individual o una empresa que accede a la plataforma para gestionar sus finanzas.
   - Se conecta a través de dos interfaces: SPA Web y Aplicación Móvil.

b. **SPA Web (Aplicación Web de Banca Digital)**
   - Es una aplicación web de una sola página (SPA - Single Page Application) que permite a los usuarios acceder a los servicios bancarios desde navegadores web.
   - Consume los servicios a través del API Gateway, asegurando una comunicación segura con la plataforma de backend.
   - Se autentica mediante OAuth 2.0 y utiliza HTTPS para garantizar la seguridad de las transacciones.

c. **Aplicación Móvil (Banca Digital Móvil)**
   - Es una aplicación multiplataforma desarrollada en Flutter o React Native que ofrece la misma funcionalidad que la SPA web pero optimizada para dispositivos móviles.

- Integra funciones avanzadas como autenticación biométrica (huella dactilar o reconocimiento facial).
- También consume los servicios a través del API Gateway.

d. API Gateway (Punto de Entrada Único del Sistema)

- Actúa como un intermediario entre las aplicaciones cliente (SPA Web y Móvil) y los microservicios en el backend.
- Funcionalidades principales:
  - Autenticación y Autorización: Maneja OAuth 2.0, autenticación multifactor (MFA) y validación de tokens.
  - Balanceo de carga: Distribuye las solicitudes entre los microservicios.
  - Gestión de tráfico y seguridad: Implementa controles de acceso, firewall de aplicaciones web (WAF) y mitigación de ataques DDoS.

e. Microservicios Bancarios (Backend del Sistema de Banca Digital)

- Grupo de microservicios desacoplados que ejecutan la lógica de negocio.
- Procesa transacciones bancarias, gestiona cuentas y permite la consulta de movimientos financieros.
- Incluye microservicios específicos como:
  - **Autenticación:** Validación de credenciales de usuario y emisión de tokens OAuth 2.0.
  - **Gestión de cuentas y movimientos:** Permite consultar saldos y movimientos en tiempo real.
  - **Procesamiento de transferencias:** Maneja pagos entre cuentas propias e interbancarias.
  - **Auditoría y cumplimiento:** Almacena registros de actividades de los usuarios.

f. Core Bancario (Sistema Central de Gestión Bancaria)

- Es el sistema centralizado del banco que almacena la información de clientes, productos, cuentas y transacciones.
- Los microservicios del backend se comunican con el Core Bancario a través de APIs o adaptadores para SOAP y REST.
- Actúa como la fuente principal de datos para operaciones bancarias críticas.

g. Plataforma de Pagos (Procesador de Pagos Externo o Interno)

- Es un sistema externo o interno que procesa pagos y transacciones interbancarias.
- Puede estar integrado con redes de pagos nacionales e internacionales para realizar transferencias a otros bancos.
- Se comunica con los microservicios bancarios para verificar disponibilidad de fondos y ejecutar pagos.

h. Sistema de Notificaciones (Envío de Alertas y Comunicaciones al Cliente)

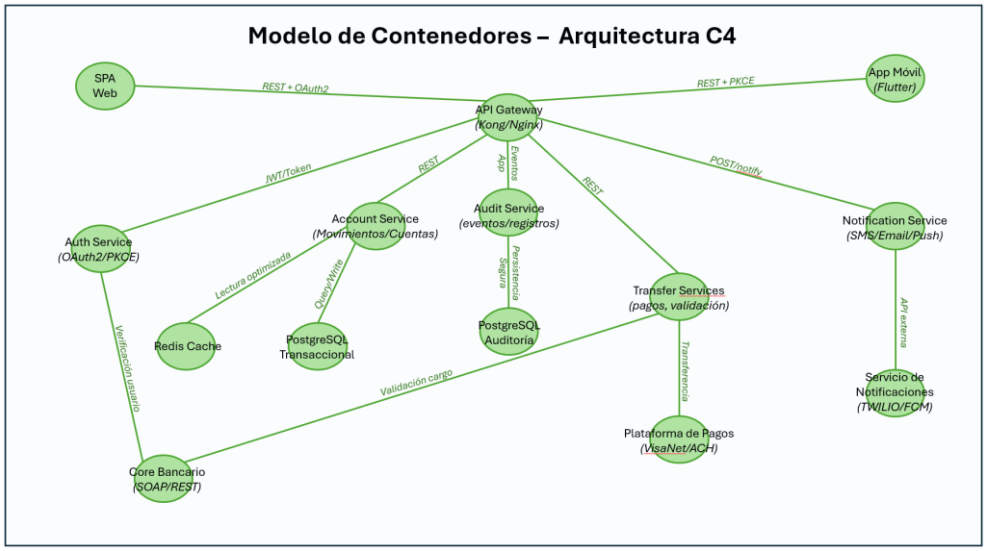
- Se encarga de enviar alertas sobre transacciones a los clientes.
- Admite múltiples canales:
  - **SMS:** A través de proveedores como Twilio o AWS SNS.
  - **Correo Electrónico:** Integración con servicios como SendGrid.
  - **Notificaciones Push:** Usando Firebase Cloud Messaging (FCM) o Apple Push Notification Service (APNS).

Flujo de Interacción entre los Componentes

1. **Inicio de sesión y autenticación:**
   - El cliente ingresa sus credenciales en la SPA Web o Aplicación Móvil.
   - Se envía la solicitud al API Gateway, que redirige al Microservicio de Autenticación.
   - Si la autenticación es exitosa, el usuario recibe un token OAuth 2.0 para futuras solicitudes.
2. **Consulta de saldos y movimientos:**
   - El usuario consulta su cuenta desde la aplicación.
   - La solicitud es procesada por el API Gateway, que la redirige al Microservicio de Movimientos.
   - El Microservicio de Movimientos obtiene la información del Core Bancario y la devuelve a la aplicación.
3. **Transferencia de dinero:**
   - El usuario ingresa los datos de la transferencia en la aplicación.
   - El API Gateway reenvía la solicitud al Microservicio de Transferencias.
   - Se verifican fondos disponibles y se envía la orden a la Plataforma de Pagos para su procesamiento.
4. **Notificación de transacción:**
   - Una vez completada la transferencia, el Microservicio de Notificaciones envía una alerta al usuario a través de SMS, correo o push.

b. Modelo de Contenedores

Este modelo detalla los componentes de software del sistema en términos de contenedores lógicos. Se representa la separación entre frontend (SPA, móvil), capa de acceso (API Gateway), microservicios de dominio (Auth, Transfer, Account, Notification, Audit), bases de datos y sistemas externos. Cada flecha muestra el tipo de comunicación entre servicios (ej: REST, POST /notify, llamadas a API externas). Se resalta cómo Transfer Service se conecta con el Core Bancario para realizar validaciones y cargos contables, y posteriormente registra los eventos en Audit Service. Se incluye uso de cache Redis y bases PostgreSQL separadas para operaciones y auditoría.



Componentes del Modelo de Contenedores:

i. Contenedor: SPA Web (Single Page Application)

Descripción: Aplicación web moderna basada en React o Angular que permite a los clientes interactuar con los servicios bancarios de forma segura y fluida.

Funciones:

- Acceso a la banca digital.
- Consulta de saldos y movimientos.
- Realización de transferencias y pagos.
- Gestión de usuarios y autenticación.

Interacción:

- Se comunica con el API Gateway para consumir los servicios del backend.
- Implementa autenticación con OAuth 2.0 y soporte para SSO (Single Sign-On).
- Utiliza HTTPS para garantizar la seguridad en la comunicación.

j. Contenedor: Aplicación Móvil

Descripción: Aplicación nativa desarrollada en Flutter o React Native, optimizada para dispositivos móviles con funcionalidades adicionales como autenticación biométrica.

Funciones:

- Experiencia de usuario adaptada a móviles.
- Acceso rápido con autenticación biométrica (huella digital, Face ID).
- Soporte para notificaciones push.

Interacción:

- Consume servicios a través del API Gateway.
- Se autentica mediante OAuth 2.0 y MFA (Autenticación Multifactor).
- Permite notificaciones en tiempo real mediante integración con Firebase Cloud Messaging (FCM).

k. Contenedor: API Gateway

Descripción: Punto de entrada único que gestiona el tráfico entre los clientes (SPA Web y Aplicación Móvil) y los microservicios del backend.

Funciones:

- Gestión de autenticación y seguridad (validación de tokens, protección contra ataques DDoS).
- Balanceo de carga y escalabilidad.
- Monitoreo y logging de solicitudes.
- Control de acceso basado en roles (RBAC).

Interacción:

- Recibe solicitudes de los clientes y las redirige al microservicio correspondiente.
- Se conecta con el servicio de autenticación para validar accesos y permisos.
- Protege los servicios internos mediante reglas de firewall y limitación de tráfico.

l. Contenedor: Microservicios Bancarios

Los microservicios son contenedores individuales que encapsulan funcionalidades específicas del sistema.

a. Microservicio de Autenticación

Funciones:

- Gestión de credenciales de usuario.
- Validación de sesiones con OAuth 2.0.
- Implementación de autenticación multifactor (MFA).

Interacción:

- Se comunica con el API Gateway para validar tokens de acceso.
- Interactúa con la base de datos de usuarios para autenticación.

b. Microservicio de Movimientos

Funciones:

- Consulta de cuentas y saldos.
- Recuperación del historial de transacciones.
- Soporte para categorización de gastos.

Interacción:

- Obtiene información en tiempo real desde el Core Bancario.
- Implementa caché con Redis para optimizar respuestas a consultas repetitivas.

c. Microservicio de Transferencias

Funciones:

- Procesamiento de transferencias entre cuentas propias e interbancarias.
- Validación de fondos disponibles.
- Aplicación de reglas antifraude.

Interacción:

- Se comunica con la Plataforma de Pagos para procesar transacciones.
- Se integra con el Sistema de Detección de Fraude basado en IA.

d. Microservicio de Notificaciones

Funciones:

- Envío de alertas a clientes por SMS, email y notificaciones push.
- Integración con plataformas como Twilio y AWS SNS.

Interacción:

- Recibe eventos desde otros microservicios (movimientos, transferencias, autenticación).
- Notifica al usuario en tiempo real sobre transacciones y eventos críticos.

e. Servicio de Auditoría

Funciones:

- Registro de eventos y acciones de usuarios en el sistema.
- Generación de reportes para cumplimiento regulatorio.

Interacción:

- Recibe eventos de todos los microservicios y almacena registros en una base de datos de auditoría.

m. Contenedor: Core Bancario

Descripción: Sistema heredado que almacena la información oficial de clientes, cuentas y transacciones.

Funciones:

- Procesa operaciones bancarias.
- Valida autenticación de usuarios en algunos procesos.
- Mantiene la contabilidad central del banco.

Interacción:

- Se comunica con los Microservicios Bancarios mediante APIs REST y adaptadores para SOAP.
- Es la fuente de verdad para los datos de cuentas y movimientos.

6. Contenedor: Plataforma de Pagos

Descripción: Sistema externo que procesa pagos interbancarios y transacciones de alto valor.

Funciones:

- Procesa transferencias y pagos electrónicos.
- Se conecta con redes de pago nacionales e internacionales.
- Aplica medidas de seguridad y validación antifraude.

Interacción:

- Recibe órdenes de pago desde el Microservicio de Transferencias.
- Notifica el estado de las transacciones al Servicio de Auditoría y al usuario.

7. Contenedor: Sistema de Notificaciones

Descripción: Servicio que gestiona el envío de alertas y notificaciones al usuario.

Funciones:

- Enviar notificaciones en múltiples canales (SMS, email, push).
- Integración con Twilio, AWS SNS y Firebase Cloud Messaging (FCM).

Interacción:

- Recibe eventos desde otros microservicios.
- Envia notificaciones a los clientes sobre movimientos y seguridad.

Relaciones entre los Contenedores

a) **Flujo de Autenticación:**
   - El usuario accede a la SPA Web o Aplicación Móvil.
   - La solicitud es enviada al API Gateway, que la redirige al Microservicio de Autenticación.
   - Si la autenticación es válida, el usuario recibe un token de acceso.

b) **Consulta de Cuentas y Movimientos:**
   - El usuario consulta su saldo y movimientos desde la aplicación.
   - La solicitud llega al API Gateway, que la redirige al Microservicio de Movimientos.
   - El Microservicio de Movimientos obtiene datos del Core Bancario y los devuelve al cliente.

c) **Ejecución de Transferencias:**
   - El usuario inicia una transferencia desde la aplicación.
   - La solicitud pasa por el API Gateway hacia el Microservicio de Transferencias.
   - Se validan los fondos disponibles y se procesa la transacción en la Plataforma de Pagos.

d) **Registro de Auditoría:**
   - Todas las operaciones importantes se registran en el Servicio de Auditoría.
   - Se generan reportes según normativas financieras.

c. Modelo de Componentes

El Modelo de Componentes en la arquitectura C4 desglosa cada uno de los microservicios en módulos específicos, detallando sus funciones y cómo interactúan dentro del ecosistema bancario. A continuación, se describen los principales componentes del sistema y su relación con otros elementos.

Este modelo descompone cada microservicio en sus módulos internos:

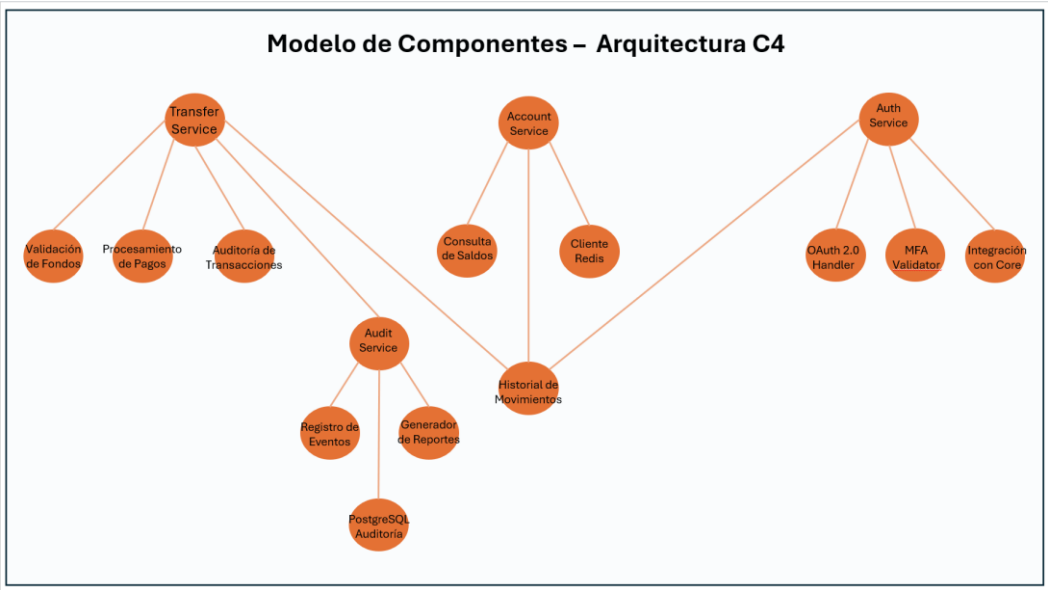
- **Transfer Service:** contiene Validación de Fondos, Procesamiento de Pagos, y Auditoría de Transacciones, e interactúa con el Core Bancario y el Audit Service.

- **Account Service:** conformado por Consulta de Saldos, Historial de Movimientos y un Cliente Redis para respuestas rápidas.

- **Auth Service:** incluye OAuth 2.0 Handler, MFA Validator, e Integración con Core para verificar datos del cliente.

- **Audit Service:** compuesto por Registro de Eventos, Generador de Reportes, y la conexión directa con PostgreSQL Auditoría.

Cada componente está orientado a una única responsabilidad, favoreciendo el mantenimiento y evolución del sistema.



n. Microservicio de Autenticación

Este microservicio es responsable de gestionar la autenticación y autorización de los usuarios en la plataforma bancaria digital.

Componentes Internos:

Módulo OAuth 2.0:

- Maneja el flujo de autenticación con estándares de seguridad basados en OAuth 2.0 y OpenID Connect.
- Emite tokens de acceso para sesiones autenticadas.
- Permite el uso de SSO (Single Sign-On) para integrar con otros servicios bancarios.

Módulo MFA (Autenticación Multifactor):

- Implementa autenticación de dos factores (2FA) mediante OTP (código de un solo uso).
- Integra autenticación biométrica (huella digital o reconocimiento facial) para la aplicación móvil.
- Soporta validaciones adicionales según el tipo de operación bancaria (por ejemplo, transferencias de alto valor).

o. Microservicio de Movimientos

Este microservicio permite a los usuarios consultar su historial de transacciones, revisar saldos y analizar gastos en tiempo real.

Componentes Internos:

Módulo de Consultas de Cuenta:

- Se conecta al Core Bancario para obtener la información de cuentas y saldos.
- Responde solicitudes en tiempo real a través de la API Gateway.
- Implementa caché con Redis para mejorar el rendimiento en consultas frecuentes.

Módulo de Historial de Transacciones:

- Registra todas las operaciones realizadas por los usuarios.
- Permite la generación de reportes de actividad en formatos PDF/Excel.
- Implementa categorización automática de gastos utilizando Machine Learning.

p. Microservicio de Transferencias

Este microservicio se encarga de procesar transferencias de dinero entre cuentas propias e interbancarias.

Componentes Internos:

Módulo de Validación de Fondos:

- Verifica la disponibilidad de fondos antes de ejecutar una transferencia.
- Aplica reglas de seguridad para evitar fraudes o transferencias sospechosas.
- Se comunica con el Sistema de Detección de Fraude basado en IA.

Módulo de Procesamiento de Pagos:

- Gestiona la ejecución de transferencias bancarias mediante la Plataforma de Pagos.
- Soporta transferencias inmediatas y programadas.
- Implementa lógica de reversión en caso de fallos durante la transacción.

q. Servicio de Auditoría

Este servicio es responsable de registrar todas las operaciones realizadas dentro del sistema para garantizar la trazabilidad y cumplimiento regulatorio.

Componentes Internos:

Módulo de Registro de Eventos:

- Registra cada acción realizada por los usuarios y administradores.
- Guarda información crítica sobre accesos, transacciones y cambios en la configuración del sistema.

Módulo de Reportes Regulatorios:

- Genera reportes de auditoría solicitados por los organismos financieros.
- Permite la consulta de registros en tiempo real con filtrado avanzado.
- Cumple con normativas como ISO 27001, PSD2 y PCI-DSS.

Relaciones entre los Componentes

Flujo de Autenticación:

a) El usuario accede a la aplicación y se autentica a través del Microservicio de Autenticación.
b) El Módulo OAuth 2.0 emite un token de sesión válido.
c) Si está activado, el Módulo MFA solicita un segundo factor de autenticación.

Consulta de Cuentas y Movimientos:

a) El usuario accede a su cuenta y solicita su saldo desde la aplicación.
b) La solicitud se dirige al Microservicio de Movimientos, que valida los datos y recupera la información desde el Core Bancario.
c) Si la consulta se ha realizado recientemente, el Módulo de Consultas de Cuenta responde desde Redis Cache para mayor rapidez.

Ejecución de Transferencias:

a) El usuario inicia una transferencia desde la aplicación.
b) El Módulo de Validación de Fondos revisa la disponibilidad de saldo.
c) Si la validación es exitosa, el Módulo de Procesamiento de Pagos ejecuta la transacción a través de la Plataforma de Pagos.
d) La transacción es registrada en el Servicio de Auditoría.

Generación de Reportes y Auditoría:

a) Todos los eventos importantes se registran en el Módulo de Registro de Eventos.
b) El Módulo de Reportes Regulatorios permite exportar registros según normativas bancarias.

11. Repositorio GitHub

Repositorio sugerido: https://github.com/<usuario>/bp-banca-digital

Estructura:

- /docs: Diagramas en .drawio/.png
- /artefactos: PDF entregable
- /README.md: Resumen, diagrama embebido, flujo de despliegue

12. Conclusión y Propuesta de Valor

La propuesta presentada resuelve de forma integral los desafíos arquitectónicos planteados para el sistema de banca digital de BP. Desde la perspectiva de un arquitecto de soluciones experto en aplicaciones empresariales, integraciones complejas y plataformas cloud, se establece una solución robusta que garantiza:

- **Interoperabilidad con sistemas legados,** preservando la inversión existente del banco.
- **Estandarización de acceso mediante OAuth 2.0 + PKCE**, mejorando la seguridad y experiencia de usuario.
- **Escalabilidad horizontal** basada en microservicios y despliegue sobre infraestructura orquestada en nube (Kubernetes, servicios gestionados).
- **Resiliencia operativa** con auditoría transaccional, mecanismos de persistencia desacoplada y servicios desacoplados vía API Gateway.
- **Cumplimiento regulatorio y monitoreo integral**, alineado a las normativas ISO 27001, GDPR y PCI-DSS.

Adicionalmente, la alineación con TOGAF y el uso del modelo C4 permiten una trazabilidad clara desde los objetivos de negocio hasta la implementación técnica.

Esta arquitectura es adaptable, segura y lista para evolucionar, permitiendo a BP acelerar su transformación digital, habilitar nuevos canales de atención y servicios financieros, y mejorar su posicionamiento competitivo en el mercado financiero digital.

Siguiente paso: consolidar los diagramas, generar los entregables PDF finales y publicar el repositorio técnico. y exportar a PDF para entrega en GitHub.

Realizado por:

ANDRES WILLIAM BANDA CELÍN

Firmado digitalmente porANDRES WILLIAM BANDA CELÍN
DN: cn=ANDRES WILLIAM BANDA CELÍN gn=ANDRES WILLIAM c=EC
Motivo:Apruebo este documento
Ubicación:
Fecha:2025-04-07 20:53-05:00

Ing. Andrés Banda C.
CC: 1713854568
Celular: 0996296875