BP – Banca por Internet

Autor: Iván Lima

Rol: Arquitecto de Soluciones

Fecha: 15/11/2025

Índice

Contenido

Resumen Ejecutivo

BP busca implementar un Sistema de Banca por Internet que permita a sus clientes consultar movimientos, realizar transferencias y pagos interbancarios, dentro de un entorno seguro, moderno y de alta disponibilidad.

La solución propuesta se basa en una arquitectura orientada a servicios, con un API Gateway como capa de integración entre los canales digitales y los sistemas internos:

- El Core Bancario provee datos de clientes, productos y movimientos.
- Un sistema complementario aporta información detallada del cliente cuando se requiere.

El sistema contará con dos canales frontales:

- Una SPA web.
- Una aplicación móvil multiplataforma.

Ambos canales se autenticarán mediante OAuth 2.0, recomendando el flujo Authorization Code con PKCE. El proceso de Onboarding móvil incorporará reconocimiento facial y autenticación biométrica para reforzar la seguridad.

Las notificaciones se enviarán por al menos dos canales (correo y SMS o push), y todas las acciones se registrarán en una base de datos de auditoría. Para optimizar el rendimiento, se aplicará un patrón de caché (Cache-Aside) con Redis para clientes frecuentes.

Esta arquitectura garantiza seguridad, escalabilidad y trazabilidad, alineándose con los objetivos de transformación digital y experiencia de usuario de BP.

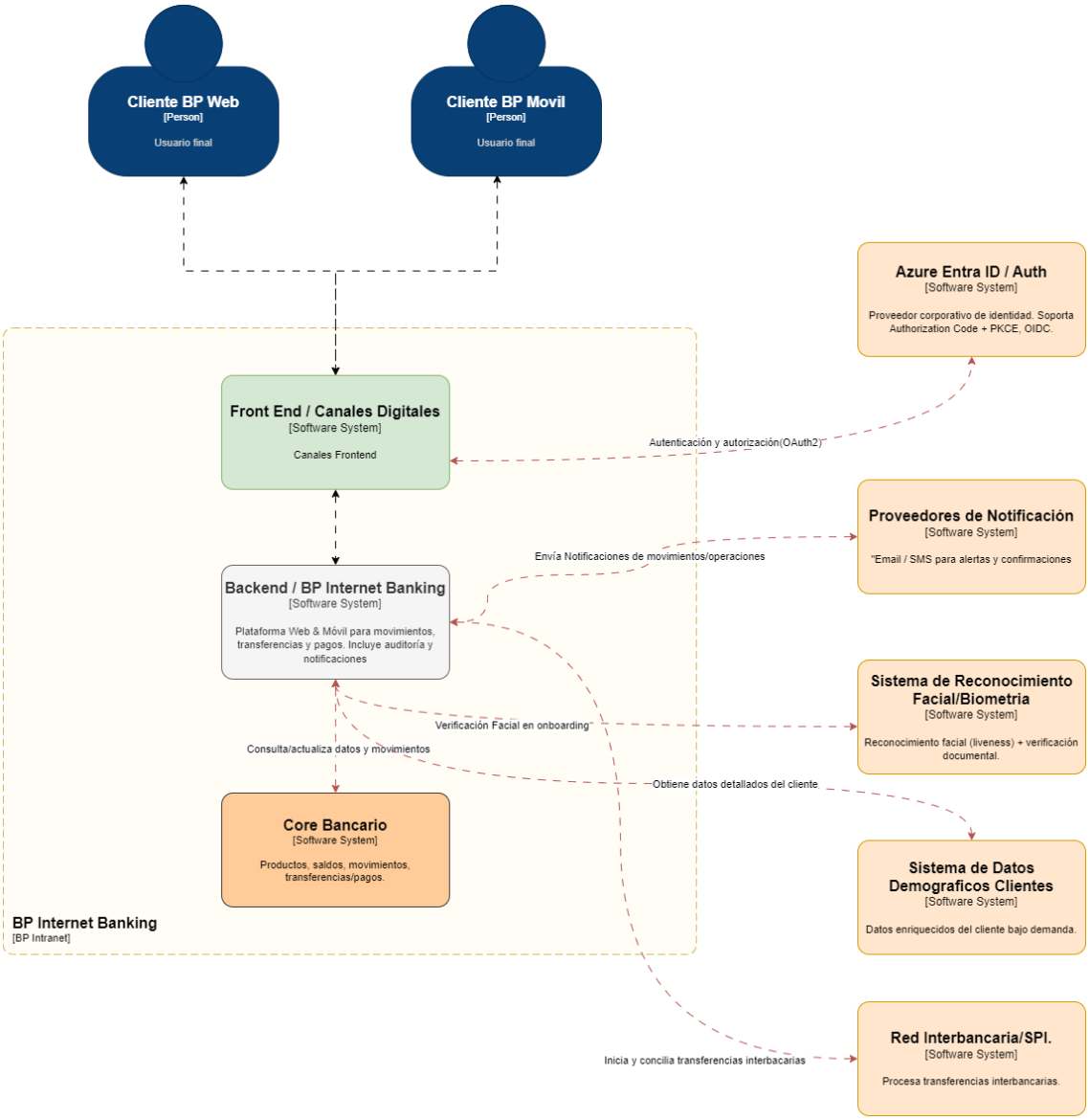
Análisis del Contexto y Actores del Sistema.

El sistema se construye sobre una arquitectura modular y escalable, integrándose con los sistemas internos y externos del banco mediante una capa de integración centralizada (API Gateway). Dicha capa permite el control de tráfico, autenticación, auditoría y orquestación de servicios, manteniendo la seguridad y consistencia de la información en todos los canales.

Descripción del Flujo General

1. El cliente accede a la **SPA Web** o a la **App Móvil**.
2. Se autentica a través del **servidor OAuth2 / IdP**, utilizando el flujo **Authorization Code con PKCE**.
3. En el caso del canal móvil, el usuario puede realizar **onboarding con verificación facial o biométrica**.
4. Una vez autenticado, el usuario interactúa con el sistema para **consultar movimientos**, **realizar transferencias** o **efectuar pagos**.
5. El **API Gateway** enruta las solicitudes hacia los servicios backend, que se comunican con el **Core Bancario** o el **Sistema Datos Demograficos**.
6. Las operaciones exitosas generan **registros en la base de auditoría** y **notificaciones automáticas** (correo/SMS).
7. En caso de transferencias interbancarias, la información es enviada y conciliada con la **Red Interbancaria / SPI**.

Modelo C4 Nivel 1 – Contexto.

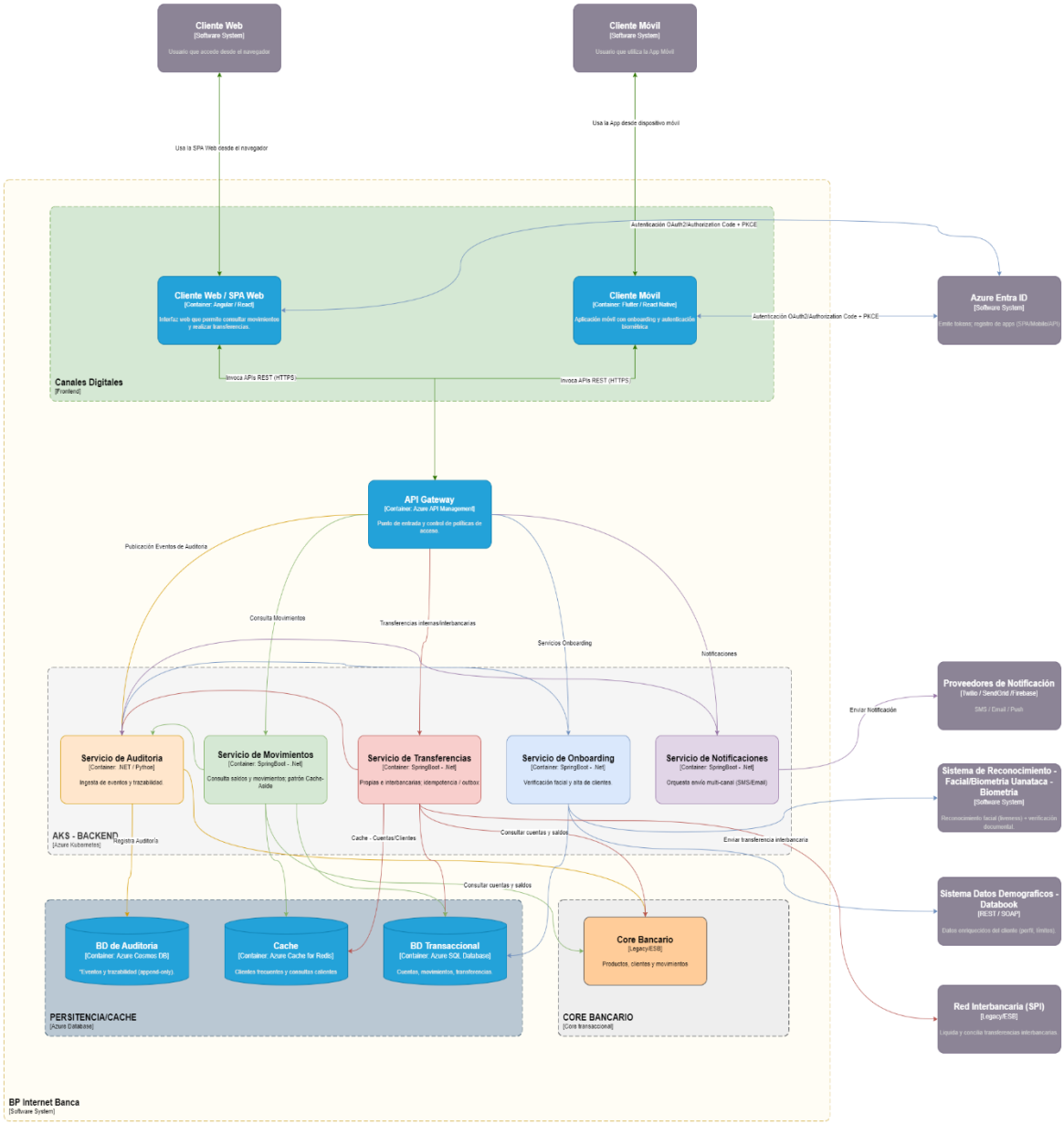


- Los **clientes** (web y móvil) son los únicos actores externos directos que interactúan con el sistema BP Internet Banking.
- El **sistema** se integra con servicios especializados del banco y externos para cumplir los procesos de autenticación, validación y operación.
- Las **interacciones** están delimitadas por el alcance del proyecto: el BP Internet Banking actúa como orquestador, mientras que la ejecución y persistencia de datos recaen en los sistemas existentes.

Actores y Sistemas Involucrados

Origen	Destino	Descripción
Cliente BP Web / Móvil	**BP Internet Banking**	Accede al sistema para realizar consultas y operaciones.
Cliente BP Web / Móvil	**OAuth2.0 / OIDC IdP**	Solicita autenticación y tokens de acceso.
BP Internet Banking	**Servicio de Datos Demográficos**	Obtiene datos enriquecidos del cliente bajo demanda.
BP Internet Banking	**Sistema de Reconocimiento Facial / Biometría**	Valida la identidad del cliente durante el onboarding.
BP Internet Banking	**Core Bancario**	Consulta o actualiza información de productos, saldos y movimientos.
BP Internet Banking	**Proveedores de Notificación**	Envía alertas y confirmaciones de operaciones.
BP Internet Banking	**Red Interbancaria / SPI**	Inicia y concilia transferencias interbancarias.

Modelo C4 – Nivel 2: Contenedores



El sistema BP Internet Banking se construye bajo una arquitectura modular basada en microservicios, desplegados en la nube (Azure) mediante Azure Kubernetes Service (AKS) y servicios PaaS complementarios. Los clientes interactúan con el sistema a través de dos canales principales: Web

(SPA) y Móvil (App Multiplataforma), ambos autenticados mediante Azure Entra ID bajo el flujo OAuth2 Authorization Code con PKCE.

Arquitectura del Canal Frontend (SPA Web & App Móvil)

La arquitectura del canal frontend está compuesta por dos aplicaciones principales: **la Web SPA** y **la aplicación móvil multiplataforma**, ambas diseñadas para ofrecer una experiencia de usuario consistente, segura y optimizada, alineada con los requerimientos funcionales del sistema de Banca por Internet.

Cliente Web (SPA)

Desarrollada en Angular o React, la SPA proporciona las funcionalidades esenciales del canal web, incluyendo consulta de saldos, movimientos, transferencias, beneficiarios, comprobantes y configuraciones del usuario. La aplicación se comunica exclusivamente a través del **API Gateway (Azure API Management)** utilizando el flujo de OAuth2 Authorization Code con PKCE, garantizando seguridad y eliminación de secretos en el lado del cliente.

Componentes clave:

- Enrutamiento seguro basado en roles y estado de autenticación.
- Auto-refresh del token de acceso y manejo de expiración.
- Integración con servicios de auditoría y notificaciones.
- Compatibilidad con navegadores modernos y accesibilidad bajo estándares WCAG.

Aplicación Móvil (Flutter / React Native)

La aplicación móvil permite un acceso rápido, seguro y con capacidades nativas para mejorar la experiencia del usuario. Incluye autenticación biométrica (FaceID, huella), notificaciones push, almacenamiento seguro de tokens y soporte para onboarding digital con captura biométrica.

Características principales:

- Integración con Azure Entra ID para autenticación OAuth2 + PKCE.
- Integración con proveedores biométricos durante el onboarding.
- Almacenamiento seguro de tokens mediante Keychain/Keystore cifrado.

- Notificaciones push mediante Firebase.
- Sincronización en tiempo real con los servicios backend para operaciones transaccionales.

Flujo de interacción Frontend – Backend

1. El usuario inicia sesión en la SPA o App Móvil.
2. El cliente solicita autenticación a Azure Entra ID mediante OAuth2 Authorization Code con PKCE.
3. Tras autenticación exitosa, el frontend recibe el **Access Token** y **ID Token**.
4. Todas las solicitudes se envían al backend únicamente a través del **API Gateway (APIM)**.
5. APIM valida el token, aplica políticas de seguridad, auditoría y versionamiento.
6. Los microservicios ejecutan la operación (consulta, transferencia, validación, etc.).
7. El resultado se devuelve al frontend y, si corresponde, se generan notificaciones y trazabilidad.

Beneficios Arquitectónicos del Frontend

- Experiencia de usuario consistente y omnicanal (web/móvil).
- Seguridad reforzada mediante PKCE, tokens de corta duración y biometría.
- Separación total del backend gracias al **enfoque API-First**.
- Despliegue independiente y capacidad de evolución sin afectar la lógica de negocio.
- Optimización del rendimiento mediante caché, estrategias de renderizado y componentes reutilizables.

Esta capa frontend complementa la arquitectura de microservicios, asegurando un flujo end-to-end completo y proporcionando la interfaz a través de la cual los clientes interactúan con todos los servicios del sistema.

APIM – API Manager

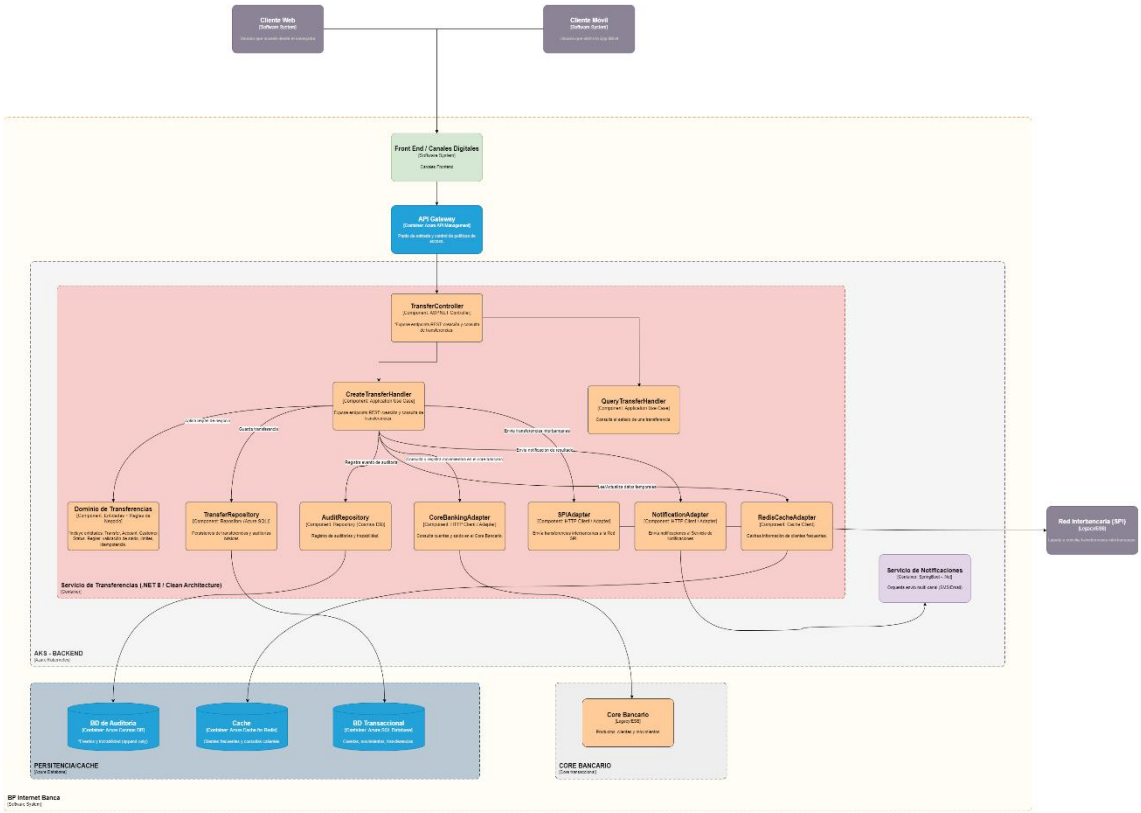
Todas las solicitudes hacia los servicios internos se enrutan a través del API Gateway (Azure API Management), que actúa como punto de entrada unificado y aplica políticas de seguridad, validación de tokens JWT, control de versiones y rate limiting.

Componentes Cloud y Contenedores Principales

Contenedor/Servicio	Tipo	Descripción breve
Cliente Web (SPA)	Aplicación Frontend (Angular/React)	Interfaz para operaciones bancarias en navegador.
Cliente Móvil	App Multiplataforma (Flutter/React Native)	Permite autenticación biométrica y acceso rápido.
Azure Entra ID	Servicio Cloud (IdP OAuth2/OIDC)	Autentica usuarios y emite tokens JWT.
API Gateway (APIM)	Servicio Cloud (API Management)	Punto de entrada seguro; valida tokens y enruta tráfico hacia backend.
Servicio de Movimientos	Microservicio (AKS / .NET 8)	Consulta saldos y movimientos de cuentas propias, conecta con core bancario.
Servicio de Transferencias	Microservicio (AKS / .NET 8)	Gestiona transferencias propias e interbancarias, conecta con core bancario.
Servicio de Auditoría	Microservicio (AKS / .NET 8)	Registra trazabilidad de operaciones y eventos normativos.
Servicio de Notificaciones externo	Microservicio (AKS / .NET 8)	Envia alertas por SMS/Email a través de proveedores externos.

Azure SQL Database	Base de datos (PaaS	Persistencia de datos transaccionales y operativos.
Azure Cosmos DB	Base de datos (PaaS)	Almacén de eventos de auditoría y logs.
Azure Cache for Redis	Base de datos (PaaS)	Acelera consultas frecuentes y reduce carga en el Core.
Servicio para Biometria (Uanataca, Adotech)	PaaS para biometría/onboarding	Realiza verificación facial durante el onboarding..
Core Bancario	Servicio Interno	Fuente principal de datos de clientes, productos y movimientos.
Servicio de Datos Demograficos (Databook, Servicios KYC)	Sistema Externo	Datos adicionales de clientes (riesgo, límites, detalle KYC, databook).
Red Interbancaria (SPI)	Sistema Externo	Procesa transferencias interbancarias.
Proveedores de Notificación	Sistema Externo	Servicios de mensajería (Twilio,Claro).

Modelo C4 – Nivel 3: Componente Servicio de Transferencias



El Servicio de Transferencias implementa los procesos de creación, validación y ejecución de transferencias internas e interbancarias.
Está diseñado bajo los principios de Clean Architecture y DDD, con separación en capas:

Capas y responsabilidades

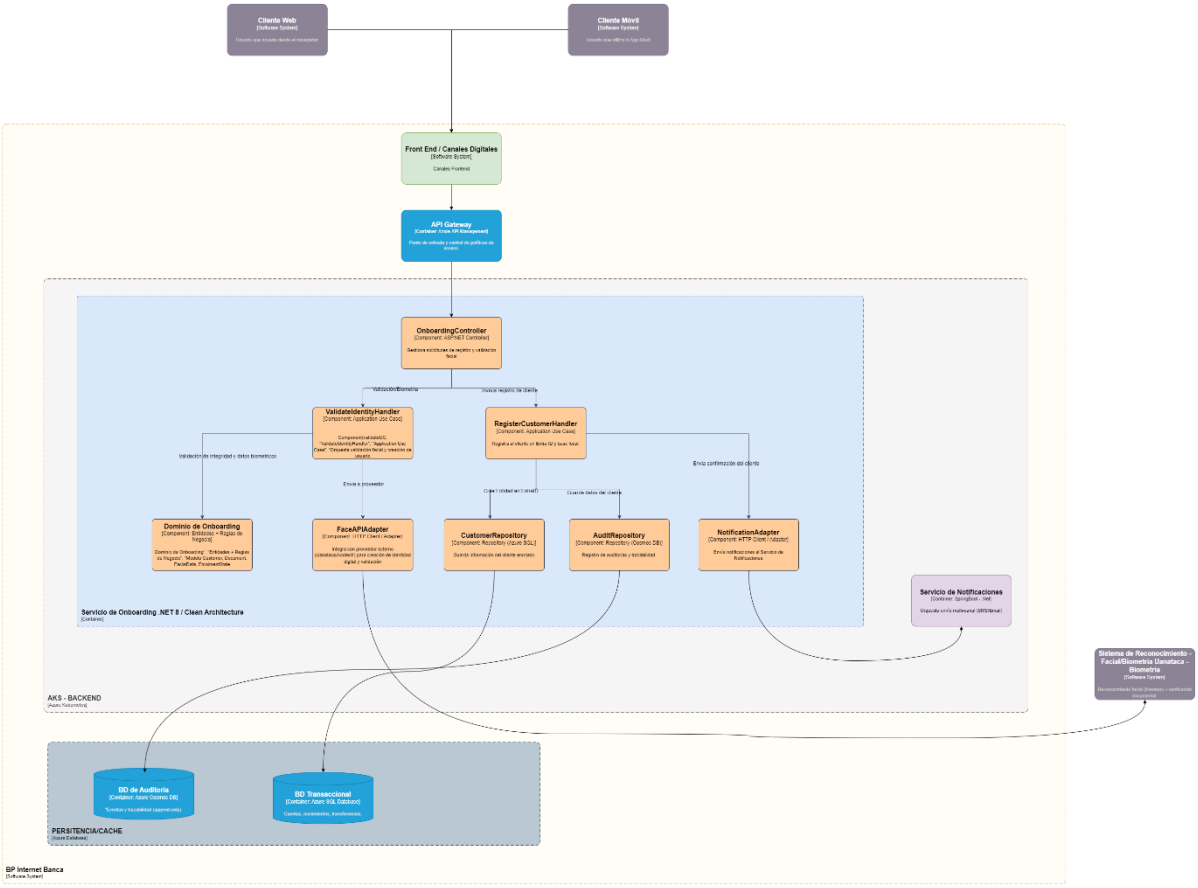
Capa	Componentes	Responsabilidad
Interface (API)	TransferController	Exponer endpoints REST, validar inputs, delegar a casos de uso.
Application (Casos de Uso)	CreateTransferHandler, QueryTransferHandler	Orquestar flujo de negocio, coordinar dominio y adaptadores.

Domain (Negocio)	Transfer, Account, Customer, Status	Representar reglas de negocio puras, sin dependencias de infraestructura.
Infrastructure (Adaptadores)	CoreBankingAdapter, SPIAdapter, NotificationAdapter, TransferRepository, AuditRepository, RedisCacheAdapter	Implementar comunicación con sistemas externos, persistencia y cacheo.

Seguridad y confiabilidad:

- Autenticación por token JWT validado en API Management.
- Idempotencia mediante control de hash y cache temporal.
- Auditoría completa en Cosmos DB.
- Secretos gestionados por Azure Key Vault.

Modelo C4 – Nivel 3: Componente Servicio de Onboarding



El Servicio de Onboarding gestiona el flujo completo de registro y validación de identidad de nuevos clientes, integrando la verificación biométrica y documental con el proveedor externo Uanataca.
Una vez validada la identidad, el servicio registra al usuario en Azure Entra ID, donde se gestionará su autenticación posterior.

Capas y responsabilidades

Capa	Componentes	Descripción
Interface (API)	OnboardingController	Exponer endpoints REST para iniciar validación de identidad y registro.
Application (Casos de uso)	ValidateIdentityHandler, RegisterCustomerHandler	Controlar el flujo de verificación con

		Uanataca y la creación del usuario en Entra ID.
Domain (Negocio)	Customer, Document, ValidationResult, EnrolmentState	Modelar la entidad del cliente, su estado de enrolamiento y las reglas de negocio.
Infrastructure (Adaptadores)	UanatacaAdapter, EntraIDAdapter, CustomerRepository, AuditRepository, NotificationAdapter	Conectarse con servicios externos, gestionar persistencia y comunicación.

Aspectos de seguridad y cumplimiento

- Los datos biométricos y documentales no se almacenan localmente, solo los resultados de validación (ValidationResult).
- Toda comunicación con Uanataca se realiza por canal cifrado TLS 1.2+ con autenticación mutua (mTLS).
- La creación del usuario en Entra ID sigue el principio Just-In-Time (JIT) tras validación exitosa.
- El proceso cumple con estándares KYC, GDPR y ISO/IEC 27001.
- Se registran todos los eventos del flujo en Cosmos DB para garantizar trazabilidad y no repudio.

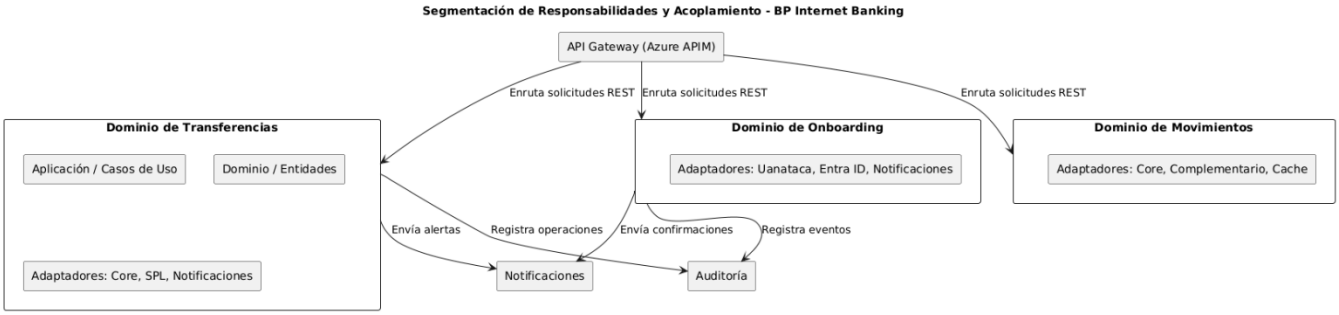
Segmentación de Responsabilidades y Acoplamiento

Principios aplicados

La arquitectura se fundamenta en los siguientes **principios de segmentación y diseño desacoplado**:

Principio	Descripción	Aplicación en el sistema

Responsabilidad única (SRP)	Cada servicio o módulo cumple una única función de negocio.	Ej. “Transferencias” gestiona operaciones monetarias; “Onboarding” valida identidad.
Separación por dominios (Bounded Contexts)	Los límites de cada dominio se definen de acuerdo con su responsabilidad funcional y sus modelos de datos.	Dominios: Onboarding, Movimientos, Transferencias, Notificaciones, Auditoría.
Desacoplamiento por capas (Clean Architecture)	Las dependencias apuntan hacia el dominio, no hacia la infraestructura.	Capa de dominio independiente del framework.
Comunicación controlada (API First)	Toda interacción entre servicios se realiza mediante contratos REST versionados.	Contratos gestionados por Azure API Management.
Evolución independiente	Cada microservicio puede desplegarse, escalar o versionarse sin afectar a los demás.	Despliegues independientes en AKS.
Acoplamiento mínimo	Los servicios se integran por interfaces bien definidas o colas/eventos.	Uso de adaptadores HTTP y repositorios desacoplados.
Cohesión alta	Las operaciones de un servicio se agrupan en torno a un mismo propósito de negocio.	Transferencias agrupa toda la lógica de orquestación de pagos.


Segmentación de Responsabilidades y Acoplamiento - BP Internet Banking
API Gateway (Azure APIM)
Enruta solicitudes REST
Enruta solicitudes REST
Enruta solicitudes REST
Dominio de Transferencias
Aplicación / Casos de Uso
Dominio / Entidades
Adaptadores: Core, SPL, Notificaciones
Dominio de Onboarding
Adaptadores: Uanataca, Entra ID, Notificaciones
Dominio de Movimientos
Adaptadores: Core, Complementario, Cache
Envía alertas
Registra operaciones
Envía confirmaciones
Registra eventos
Notificaciones
Auditoría

Justificación de Decisiones Arquitectónicas.

A continuación, se detallan las principales decisiones y sus justificaciones teóricas:

Decisión Arquitectónica	Justificación Teórica
Uso de Clean Architecture	Separa claramente las capas de dominio, aplicación e infraestructura, reduciendo el acoplamiento entre lógica de negocio y frameworks. Facilita la mantenibilidad, pruebas unitarias y reemplazo de dependencias sin afectar la lógica central.
Aplicación de DDD (Domain Driven Design)	Permite modelar los dominios de negocio (Onboarding, Transferencias, Movimientos) con límites funcionales claros (Bounded Contexts). Mejora la comprensión del negocio y la alineación entre equipos técnicos y funcionales.
Despliegue en Azure Kubernetes Service (AKS)	Proporciona escalabilidad automática (Horizontal Pod Autoscaler) y alta disponibilidad. Facilita CI/CD y observabilidad mediante integración nativa con Azure Monitor y Application Insights.
Uso de Azure API Management como Gateway	Centraliza políticas de seguridad, control de acceso y versionamiento de APIs. Permite auditoría y trazabilidad de consumo, cumpliendo con estándares regulatorios.
Integración con Uanataca para validación de identidad	Uanataca es un proveedor certificado con cumplimiento KYC y normativas eIDAS, adecuado para el contexto bancario. Ofrece validación facial y documental robusta, reduciendo riesgo de fraude.
Autenticación con Azure Entra ID (OAuth2 / OIDC)	Estándar ampliamente adoptado y compatible con flujos seguros como Authorization Code + PKCE. Permite gestión centralizada de identidades, MFA y control granular de acceso (RBAC).

Persistencia en Azure SQL + Cosmos DB	SQL garantiza integridad y consistencia transaccional (ACID). Cosmos DB provee escalabilidad horizontal y baja latencia para auditorías y trazabilidad.
Cache con Azure Redis (Cache-Aside Pattern)	Mejora el rendimiento al reducir llamadas al Core Bancario en operaciones frecuentes. Implementa un patrón de diseño probado que mantiene consistencia eventual entre caché y base de datos.
Uso de patrones Repository y Adapter	Desacoplan la lógica del dominio de los detalles de infraestructura o protocolos externos. Facilitan pruebas unitarias y reemplazo de dependencias sin cambios en el dominio.
Protocolos REST + JSON	Estándar ampliamente compatible y fácil de versionar en entornos heterogéneos. Permite interoperabilidad entre sistemas internos y externos (Core, SPL, Uanataca).

Alternativas Evaluadas.

Durante el diseño del sistema BP Internet Banking, se evaluaron diversas alternativas tecnológicas y de patrones arquitectónicos.
Se eligieron principalmente servicios nativos de Microsoft Azure, priorizando su integración fluida, soporte empresarial, seguridad certificada y facilidad de despliegue automatizado mediante IaC (Terraform).

La elección de Azure permite gestionar infraestructura, identidades, APIs y bases de datos dentro de un mismo ecosistema, reduciendo la complejidad operativa y el tiempo de aprovisionamiento.
No obstante, el diseño es portátil y agnóstico de proveedor, ya que todos los componentes siguen estándares abiertos (OAuth2, REST, Docker, Kubernetes, Terraform), lo que facilita una futura migración a otras nubes como AWS (EKS, Cognito, RDS) o GCP (GKE, IAM, Cloud SQL) sin reescribir la lógica de negocio.

Componente / Decisión	Alternativas Evaluadas	Elección Final	Razonamiento Resumido
Plataforma Cloud	AWS / GCP / Azure	**Azure**	Integración nativa entre servicios, compliance financiero y despliegue IaC con Terraform.
Orquestación de contenedores	AWS EKS / GKE / AKS	**AKS**	Servicio administrado, integración con ACR, autoscaling y monitoreo nativo.
API Gateway	Kong / NGINX / Azure API Management	**APIM**	Seguridad OAuth2, versionamiento y analítica integrada.
Identidad y autenticación	Keycloak / Auth0 / Azure Entra ID	**Entra ID**	OIDC, MFA y políticas centralizadas.
Persistencia	PostgreSQL / MongoDB / Cosmos DB + Azure SQL	**Azure SQL + Cosmos DB**	Cumplimiento ACID + auditoría escalable.
Validación biométrica	Face API / Jumio / Uanataca	**Uanataca**	Cumple eIDAS/KYC, mayor respaldo regulatorio.

Patrones de Arquitectura Aplicados.

El sistema BP Internet Banking aplica un conjunto de patrones arquitectónicos que garantizan modularidad, escalabilidad y mantenibilidad, alineados con prácticas modernas para entornos financieros y cloud-native.

Patrón	Propósito / Aplicación
Clean Architecture	Separa la lógica de negocio de la infraestructura, organizando cada servicio en capas (*Domain*, *Application*, *Infrastructure*, *API*).
Domain-Driven Design (DDD)	Define límites funcionales claros por dominio (Onboarding, Transferencias, Auditoría), asegurando cohesión y bajo acoplamiento.
Ports and Adapters	Aísla la lógica central de dependencias externas (Core, SPL, Uanataca, Notificaciones) mediante interfaces y adaptadores.
Repository Pattern	Centraliza el acceso a datos en Azure SQL y Cosmos DB, ocultando detalles de persistencia.
Cache-Aside Pattern	Usa **Azure Redis** para reducir la carga en el Core Bancario durante consultas frecuentes.
API First	Las APIs se diseñan antes del desarrollo y se gestionan mediante **Azure API Management** con versionamiento y políticas de seguridad.

Beneficios:
Arquitectura mantenible y extensible, desacoplada de proveedores, preparada para automatización IaC con Terraform y adaptable a otros entornos (AWS, GCP).

Integración con Servicios Externos.

El sistema BP Internet Banking interactúa con múltiples servicios externos que proporcionan información y funcionalidades críticas para la operación bancaria. Todas las integraciones se realizan a través del API Gateway (Azure API Management), el cual valida tokens OAuth2 emitidos por Azure Entra ID y aplica políticas de seguridad, límite de consumo y registro de auditoría.

Servicio Externo	Propósito	Tipo de Integración
Core Bancario	Proporciona información de cuentas, movimientos y saldos.	REST con autenticación mTLS.
Sistema Complementario	Entrega información extendida de cliente (riesgo, límites, KYC).	REST seguro vía API Gateway.
Red Interbancaria (SPI)	Procesa transferencias hacia otros bancos.	REST síncrono + confirmación de respuesta.
Uanataca	Valida identidad y biometría de nuevos clientes durante el onboarding.	REST asíncrono con retorno de resultado validado.
Proveedores de Notificaciones, firebase	Envía SMS y correos electrónicos al cliente.	REST/SMTP gestionado por el microservicio de Notificaciones.

Beneficio arquitectónico

Esta estrategia de integración garantiza:

- **Bajo acoplamiento** entre servicios internos y proveedores externos.
- **Trazabilidad total** mediante auditoría en **Cosmos DB**.
- **Portabilidad**: al basarse en APIs estándar REST y contratos OpenAPI, los adaptadores pueden migrarse a otros entornos (*AWS API Gateway*, *GCP Apigee*) sin modificar la lógica del dominio.

Arquitectura de Autenticación y Autorización

La autenticación del sistema BP Internet Banking se basa en el estándar OAuth2.0 / OpenID Connect (OIDC), utilizando Azure Entra ID como proveedor de identidad (IdP).
El modelo centraliza la gestión de credenciales, tokens y políticas de acceso, garantizando seguridad, trazabilidad y cumplimiento con normativas financieras.

Componente	Función
Azure Entra ID (IdP)	Gestiona la autenticación de usuarios, emisión de tokens de acceso (JWT) y refresh tokens.
API Gateway (APIM)	Valida los tokens en cada solicitud y aplica políticas de autorización por servicio.
Aplicaciones Front (SPA / Móvil)	Implementan el flujo **Authorization Code con PKCE**, adecuado para clientes públicos sin secreto.
Microservicios internos	Consumen tokens validados por APIM, aplicando reglas de acceso específicas según el rol o contexto.

Flujo de autenticación

1. El usuario inicia sesión desde la **app móvil** o la **SPA web**.
2. El cliente solicita autenticación a **Azure Entra ID**, que presenta opciones de ingreso (clave, huella, FaceID o token MFA).
3. Tras autenticación exitosa, **Entra ID** devuelve un **authorization code**.
4. El cliente intercambia ese código por un **token JWT** firmado (Access Token + ID Token).
5. El **API Gateway (APIM)** valida el token, extrae claims y enruta la solicitud al microservicio correspondiente.
6. Los microservicios confirman el contexto del usuario (rol, permisos, cliente) sin necesidad de credenciales locales.

Relación con Onboarding (Uanataca)

- El **proceso de Onboarding** integra validación facial y documental a través de **Uanataca**, garantizando la identidad del usuario antes de crear su cuenta en **Azure Entra ID**.
- Una vez validado, el usuario queda enrolado en Entra ID y puede autenticarse mediante los métodos soportados: **usuario/clave**, **biometría (FaceID/huella)** o **MFA corporativo**.
- El token emitido por Entra ID se usa posteriormente para acceder a todos los servicios internos del sistema.

Seguridad y cumplimiento

- Flujos certificados bajo **OAuth2 + OIDC**, con validación de firmas JWT (RS256).
- Tokens de corta duración y refresco controlado.
- Gestión de secretos mediante **Azure Key Vault**.
- Cumplimiento con normativas **PSD2**, **ISO 27001** y lineamientos de **seguridad bancaria local**.

Beneficio arquitectónico

Este modelo desacopla la autenticación del backend, mejora la experiencia del usuario y permite aplicar políticas centralizadas de acceso. Además, al utilizar **estándares abiertos y Azure Entra ID**, la solución puede migrarse a otros IdP (Auth0, Keycloak, AWS Cognito) sin afectar la lógica de negocio.

Arquitectura de Onboarding y Validación de Identidad

El proceso de Onboarding permite registrar y verificar la identidad de nuevos clientes antes de habilitar su acceso al sistema BP Internet Banking. Este flujo combina validación biométrica y documental mediante Uanataca, la creación segura de la identidad digital en Azure Entra ID, y la auditoría completa de cada evento.

Componente	**Función**
App Móvil (Onboarding UI)	Captura foto facial, documento y metadatos del cliente.

Servicio de Onboarding (.NET / AKS)	Orquesta el proceso de validación, recibe los datos y los envía a Uanataca.
Uanataca API	Valida la coincidencia facial y la autenticidad del documento (KYC + biometría).
Azure Entra ID	Crea la identidad digital del cliente validado, habilitando métodos de ingreso seguros.
Auditoría (Cosmos DB)	Registra el resultado y trazabilidad de cada intento de enrolamiento.
Notificaciones firebase	Informa al usuario sobre el resultado del proceso.

Flujo del proceso de Onboarding

1. El usuario inicia el registro desde la **aplicación móvil**.
2. La app envía la solicitud con datos biométricos y documentos al **Servicio de Onboarding**.
3. El servicio envía la información a **Uanataca**, que realiza la validación facial y documental.
4. Si la validación es exitosa, el sistema crea la identidad del usuario en **Azure Entra ID**.
5. Se almacena la trazabilidad del evento en **Cosmos DB** (fecha, resultado, origen, dispositivo).
6. Se notifica al cliente por SMS o correo sobre el estado del proceso.

Controles de seguridad

- **Cifrado de extremo a extremo (TLS 1.2+)** en todo el flujo.
- **No se almacenan imágenes biométricas completas**, solo vectores de validación o resultados hash.
- **Trazabilidad completa** del proceso en **Cosmos DB** para auditorías regulatorias.
- **Tokens temporales y secretos protegidos** en **Azure Key Vault**.

Beneficio arquitectónico

Este modelo garantiza:

- Alta **confiabilidad en la validación de identidad** (biometría + documento).
- Cumplimiento con estándares **eIDAS**, **KYC** y **ISO/IEC 27001**.
- **Desacoplamiento del proveedor** (Uanataca puede ser reemplazado sin afectar el dominio).
- Integración directa con el flujo de autenticación de **Azure Entra ID** para acceso inmediato.

Auditoría y Trazabilidad

Toda la información se almacena en un repositorio independiente basado en **Azure Cosmos DB**, diseñado para alta disponibilidad, baja latencia y resiliencia ante fallos.

Componente	Función principal
Servicio de Auditoría	Recibe los eventos generados por los microservicios internos y los almacena en Cosmos DB.
Cosmos DB (PaaS)	Base de datos NoSQL que almacena registros estructurados y eventos normativos.
Azure Key Vault	Protege las credenciales y cadenas de conexión utilizadas en el servicio de auditoría.
Application Insights / Monitor	Recolecta métricas operativas, latencias y errores para observabilidad completa.

Flujo de auditoría

1. Cada microservicio (Transferencias, Onboarding, Movimientos) envía un **evento de auditoría** al servicio central.
2. El evento contiene información como: usuario, fecha, acción, canal, resultado, dirección IP, correlación y nivel de criticidad.
3. El servicio normaliza y almacena los registros en **Cosmos DB**.

4. Las herramientas de monitoreo y cumplimiento pueden consultar estos eventos en tiempo real o histórico.

Controles y cumplimiento

- **Inmutabilidad:** los registros no pueden modificarse, solo agregarse nuevos eventos.
- **Disponibilidad:** Cosmos DB ofrece SLA del 99.999% y replicación geográfica.
- **Integridad:** los datos son cifrados en tránsito y en reposo (AES-256).
- **Cumplimiento:** se cumple con normativas **ISO 27001**, **PCI DSS**.

Beneficio arquitectónico

Este modelo garantiza:

- **Trazabilidad total** de cada operación y cambio de estado.
- **Centralización del registro** para facilitar auditorías internas o regulatorias.
- **Desacoplamiento**: los servicios solo publican eventos, sin lógica dependiente del almacenamiento.
- **Escalabilidad horizontal**, al permitir procesar grandes volúmenes de eventos simultáneamente.

Alta Disponibilidad y Monitoreo

Se aplican estrategias de alta disponibilidad tanto a nivel de infraestructura como de aplicación, utilizando servicios nativos de Azure y prácticas DevOps.

Mecanismos de alta disponibilidad:

Componente / Nivel	Mecanismo aplicado	Objetivo

Azure Kubernetes Service (AKS)	Clúster multi-nodo con *auto-healing* y *Horizontal Pod Autoscaler*.	Garantizar continuidad y escalabilidad automática según carga.
Bases de datos (Azure SQL / Cosmos DB)	Replicación geográfica activa y failover automático.	Asegurar disponibilidad de datos y recuperación ante desastres.
API Management (APIM)	Instancias distribuidas y *regional gateway*.	Evitar puntos únicos de fallo en el acceso a servicios.
Redis Cache	Redundancia primaria/secundaria y sincronización automática.	Evitar pérdida de sesiones o datos temporales.
Contenedores Docker	*Rolling updates* y *liveness/readiness probes*.	Mantener disponibilidad durante despliegues o reinicios.

Monitoreo y observabilidad:

Herramienta / Servicio	Uso principal
Azure Monitor / Log Analytics	Centraliza métricas, logs y alertas en tiempo real.
Application Insights	Registra trazas, dependencias y rendimiento de cada microservicio.
Container Insights (AKS)	Supervisa recursos, pods, CPU, memoria y eventos del clúster.
Dashboards personalizados (Grafana / Power BI)	Visualización ejecutiva de KPIs y disponibilidad del sistema.

Gestión de incidentes y alertas

- Alertas automáticas configuradas por umbral (CPU, latencia, errores HTTP 5xx).
- Integración con canales de soporte (Teams / correo / PagerDuty) para respuesta inmediata.
- Logs estructurados y correlacionados mediante RequestId para trazabilidad cruzada entre servicios.

Beneficio arquitectónico

Esta estrategia garantiza:

- Disponibilidad superior al 99.95%.
- Escalado automático ante picos de carga.
- Visibilidad completa del estado del sistema en tiempo real.
- Capacidad de reacción rápida ante anomalías operativas o de seguridad.

Plan de Continuidad Operativa y Contingencia

Para garantizar la operación ininterrumpida del sistema de Banca por Internet y reducir el riesgo ante eventos disruptivos, la arquitectura incorpora un conjunto de medidas orientadas a la continuidad del servicio y la recuperación ante desastres (BCP/DRP).

1. Estrategia de Continuidad Operativa (BCP)

- **Componentes críticos distribuidos** en múltiples zonas de disponibilidad de Azure.
- **Auto-escalado horizontal** para absorber picos de demanda.
- **Health checks** y mecanismos de auto-recuperación en AKS.
- **APIM con fallback por regiones**, permitiendo enrutar el tráfico en caso de falla local.
- **Caché Redis replicado**, evitando pérdida de sesiones o datos temporales.
- **Persistencia transaccional** con recursos siempre-on en Azure SQL.

2. Recuperación ante Desastres (DRP)

Se implementa un enfoque multi-región y multi-copia:

- **Azure SQL geo-replicado** con failover automático.

- **Cosmos DB con replicación geográfica activa**, permitiendo lectura/escritura regional.
- **Backup automático** en todas las bases de datos (retención 14–30 días).

3. Procedimientos de Contingencia

- **Modo degradado** para funcionalidades no críticas (ej. consultas básicas, notificaciones).
- **Circuit breakers** entre microservicios para evitar cascadas de fallas.
- **Políticas de reintento** y colas de compensación en transferencias.
- **Failover manual** del API Gateway si la región primaria presenta incidentes.
- **Plan de operación manual** para conciliación offline de transacciones en casos extremos.

4. Ejecución del Plan y Pruebas

- Ensayos **semestrales** del plan de recuperación ante desastres.
- Auditoría de cumplimiento y validación de RTO/RPO:
    - **RTO (Recovery Time Objective):** 15–30 minutos
    - **RPO (Recovery Point Objective):** 0–5 minutos
- Integración con procesos internos de continuidad del negocio del banco.

5. Beneficios Operativos

- Garantía de disponibilidad superior al 99.95%.
- Reducción de tiempos de interrupción ante fallos críticos.
- Seguridad y resiliencia alineadas con las normativas del sector financiero.
- Capacidad de recuperación completa con mínima pérdida de datos.

Regulaciones Bancarias y Estándares de Seguridad

La arquitectura fue diseñada bajo principios de **Zero Trust**, **cifrado extremo a extremo** y **seguridad por diseño**, aplicando controles tanto en la capa de infraestructura como en la lógica de aplicación.

Controles de seguridad implementados:

Área de Seguridad	Controles Aplicados
Autenticación y Autorización	Basada en **OAuth2 / OpenID Connect** con **Azure Entra ID**, MFA y tokens JWT firmados (RS256).
Protección de Datos	Cifrado en tránsito (**TLS 1.2+**) y en reposo (**AES-256**). Datos sensibles aislados por dominio.
Gestión de Credenciales	Secretos, certificados y tokens almacenados en **Azure Key Vault**.
Auditoría y No Repudio	Registro de eventos en **Cosmos DB**, inmutables y consultables por entes regulatorios.
Seguridad en APIs	Validación de tokens, *rate limiting* y *IP filtering* mediante **API Management**.
Infraestructura Segura (AKS)	Políticas de *network segmentation*, *private endpoints*, *RBAC* y *Azure Defender for Containers*.
Despliegue Seguro (DevSecOps)	Análisis de vulnerabilidades en CI/CD, imágenes firmadas y escaneo de dependencias.

Cumplimiento normativo

El diseño y operación cumplen con los principales marcos regulatorios y estándares del sector financiero:

- **ISO/IEC 27001:** Sistema de Gestión de Seguridad de la Información.
- **PCI DSS:** Manejo seguro de datos de pago.
- **GDPR / Ley de Protección de Datos Personales:** Tratamiento responsable de información sensible.

- **PSD2:** Autenticación reforzada (SCA) y autorización segura de operaciones financieras.

PYC (Prevención y Control de Lavado de Activos y Financiamiento del Terrorismo):

- Validación de identidad reforzada en el Onboarding con Uanataca (KYC).
- Registro de operaciones inusuales en el módulo de Auditoría.
- Integración con sistemas de análisis de riesgo y monitoreo transaccional.

Beneficio arquitectónico

Esta estructura de seguridad garantiza:

- **Confidencialidad y privacidad** de la información del cliente.
- **Integridad y disponibilidad** de las operaciones bancarias.
- **Cumplimiento regulatorio continuo**, con auditorías internas y externas.
- **Capacidad de adaptación multicloud**, manteniendo el mismo modelo de seguridad mediante políticas y herramientas equivalentes (ej. AWS KMS, GCP Secret Manager).

Modelo de Costos y Consideraciones Presupuestarias.

La propuesta arquitectónica debe dimensionarse para operar bajo los estándares de una institución financiera de alto volumen, considerando sus niveles de transaccionalidad, cargas concurrentes, criticidad regulatoria y demanda de disponibilidad.

Para ello, los costos y capacidades se proyectan bajo un modelo **enterprise**, preparado para sostener picos de carga, escalamiento agresivo y continuidad operativa 24/7.

1. Escenarios de Volumen y Tendencias de Uso en Banca Digital

Como referencia, se asumen:

- **5,000 – 12,000 solicitudes por minuto** en horas pico (SPA + App Móvil).
- **1M – 2.5M solicitudes diarias**.

- **80% consultas** (saldos, movimientos, beneficiarios).
- **20% transacciones críticas** (transferencias, pagos interbancarios, validaciones).
- **Disponibilidad requerida: ≥ 99.98%** (tiempo máximo de caída permitido: ~9 minutos mensuales).
- **RTO:** 5–15 minutos
- **RPO:** 0–5 minutos

Estos parámetros definen la capacidad mínima de infraestructura para un entorno productivo corporativo.

Para cubrir estos niveles de carga, se recomienda un enfoque empresarial basado en servicios administrados de Azure:

- Un clúster de AKS entre 6 y 9 nodos de producción, con capacidad de autoescalado hasta 12 nodos durante los picos.
- Azure API Management en plan Premium, habilitado en al menos dos regiones para garantizar redundancia activo-activo.
- Azure SQL Database en modalidad Business Critical con entre 8 y 16 vCores, permitiendo un volumen alto de consultas transaccionales.
- Cosmos DB configurado con 30.000 a 50.000 RU/s, incluyendo replicación geográfica para auditoría y registro de eventos.
- Redis Enterprise Premium como caché de alto rendimiento, capaz de sostener más de 100.000 operaciones por segundo y reducir la presión sobre el Core Bancario.
- Servicios complementarios como Log Analytics, Application Insights, Key Vault y Azure Storage para telemetría, seguridad y almacenamiento de evidencia.

Inversión Inicial del Proyecto (CAPEX)

La solución requiere una inversión inicial destinada a la construcción, integración y puesta en marcha de la plataforma.

El CAPEX incluye:

- Desarrollo de los microservicios de Transferencias, Movimientos, Onboarding, Auditoría y Notificaciones.

- Construcción del canal Web (SPA) y de la Aplicación Móvil.
- Configuración del clúster AKS, API Management Premium, SQL Business Critical, Cosmos DB y Redis Enterprise.
- Implementación de infraestructura como código (IaC), pipelines CI/CD y automatización de despliegues.
- Integraciones con Core Bancario, Red SPI, proveedores de biometría y servicios externos.
- Hardening de seguridad, pruebas técnicas, pruebas de carga y certificaciones.
- Preparación de ambientes QA, Preproducción y Producción.

CAPEX estimado:

Entre 60.000 y 120.000 USD, dependiendo del alcance final del proyecto, el número de funcionalidades incluidas y la complejidad del onboarding biométrico e integraciones.

Estimación de costos mensuales (OPEX)

El costo mensual estimado para operar esta plataforma de forma estable en un entorno productivo de alto volumen se encuentra en un rango entre **13.000 y 21.000 USD**. Este valor incluye el clúster de AKS, API Management Premium, bases de datos Azure SQL y Cosmos DB, Redis Enterprise, monitoreo centralizado, almacenamiento, secretos y componentes complementarios necesarios para cumplimiento y resiliencia.

Este rango puede ajustarse dependiendo de variables como:

- Número real de solicitudes por minuto.
- Volumen de operaciones interbancarias.
- Cantidad de validaciones biométricas.
- Uso de RU/s de Cosmos DB.
- Frecuencia de picos de carga.

Componente	**Tier / Capacidad**	**Costo estimado mensual**

AKS (6–12 nodos)	Nodos D4as v5 + autoscaling	**$4,000 – $7,500**
APIM Premium	Multi-región, throughput alto	**$3,500 – $5,000**
Azure SQL BC	8–16 vCores	**$2,500 – $4,000**
Cosmos DB	30K–50K RU/s	**$1,500 – $2,500**
Redis Enterprise Premium	100K ops/s	**$800 – $1,200**
Log Analytics + App Insights	Telemetría alta	**$700 – $1,200**
Storage + Key Vault	Evidencias, respaldos, secretos	**$100 – $250**

El dimensionamiento se adapta automáticamente mediante escalamiento horizontal en AKS, incremento de unidades en API Management y aumento dinámico de throughput en Cosmos DB. De esta manera, cuando el tráfico crezca a 15.000 o 20.000 solicitudes por minuto, la plataforma podrá absorber el incremento sin necesidad de rediseños estructurales.
Cada componente escalará en función de la carga:

- El clúster AKS aumentará nodos y réplicas de pods.
- API Management agregará instancias Premium según el throughput requerido.
- Cosmos DB ampliará automáticamente las RU/s.
- Redis soportará mayor volumen sin afectar la latencia.

RESUMEN OPEX vs CAPEX

Concepto	**Costo Estimado**	**Descripción**
CAPEX (Inversión Inicial)	USD 60.000 – USD 120.000	Desarrollo, integración, infraestructura inicial, IaC, CI/CD, seguridad, pruebas y certificación.
OPEX (Costo Operativo Mensual)	USD 13.000 – USD 21.000 / mes	Consumo mensual de recursos cloud y servicios administrados para operación del canal digital.

Conclusión

El diseño arquitectónico del sistema **BP Internet Banking** cumple integralmente con los requerimientos funcionales y no funcionales planteados, proponiendo una solución **segura, escalable, modular y alineada con los estándares de la industria financiera**.

La propuesta se fundamenta en principios modernos de arquitectura —**Clean Architecture**, **Domain-Driven Design (DDD)**, **API First** y **Ports & Adapters**— que aseguran un bajo acoplamiento entre componentes, alta cohesión dentro de cada dominio y facilidad de evolución tecnológica a futuro.

Síntesis de la solución propuesta

- **Arquitectura basada en microservicios** desplegada en **Azure Kubernetes Service (AKS)**, con gestión centralizada a través de **API Management**.
- **Integración segura** con sistemas externos: **Core Bancario**, **SPL**, **Uanataca**, y proveedores de **Notificaciones**.
- **Autenticación unificada** mediante **Azure Entra ID (OAuth2 + PKCE)** y **Onboarding biométrico con Uanataca**, garantizando cumplimiento **KYC / PYC**.
- **Auditoría y trazabilidad centralizadas** en **Azure Cosmos DB**, con registros inmutables y cumplimiento normativo.
- **Alta disponibilidad y monitoreo continuo**, con observabilidad completa en **Application Insights** y **Log Analytics**.
- **Cumplimiento integral** con estándares **ISO 27001, PCI DSS, GDPR, PSD2** y normativas locales de **Superintendencia de Bancos** y **Prevención de Lavado de Activos (PYC)**.

Beneficios del diseño

- **Escalabilidad horizontal** y despliegue automatizado mediante **IaC (Terraform)**.
- **Seguridad integral**, desde la autenticación hasta el almacenamiento de datos.

- **Portabilidad multicloud**, posible de trasladar a **AWS, GCP u Oracle Cloud** sin cambios funcionales significativos.
- **Sostenibilidad tecnológica**, gracias al uso de patrones arquitectónicos probados y servicios gestionados (PaaS).