# PROPUESTA EJERCICIO PRACTICO

## PROPUESTA DE TECNICA

**Preparado por:**

Christian Yépez
christyepez@gmail.com

# TABLA DE CONTENIDO

# ANTECEDENTES

Diseñar una arquitectura de implementación sistema Bancario

Escenario:

Usted ha sido contratado por una entidad llamada BP como arquitecto de soluciones para diseñar un sistema de banca por internet, en este sistema los usuarios podrán acceder al histórico de sus movimientos, realizar transferencias y pagos entre cuentas propias e interbancarias.

Toda la información referente al cliente se tomará de 2 sistemas, una plataforma Core que contiene información básica de cliente, productos, cuentas, movimientos, y un sistema independiente que complementa la información del cliente cuando los datos se requieren en detalle este sistema core está conectado a una base de datos principal.

Debido a que la norma exige que los usuarios sean notificados sobre los movimientos realizados, el sistema utilizará sistemas externos o propios de envío de notificaciones, mínimo 2.

Este sistema contará con 2 aplicaciones en el Front, una SPA y una Aplicación móvil desarrollada en un Framework multiplataforma. (Mencione 2 opciones y justifique el porqué de su elección).

Ambas aplicaciones autenticarán a los usuarios mediante un servicio que usa el estándar Oauth2.0, para el cual no requiere implementar toda la lógica, ya que la compañía cuenta con un producto que puede ser configurado para este fin; sin embargo, debe dar recomendaciones sobre cuál es el mejor flujo de autenticación que se debería usar según el estándar.

Tenga en cuenta que el sistema de Onboarding para nuevos clientes en la aplicación móvil usa reconocimiento facial, por tanto, su arquitectura deberá considerarlo como parte del flujo de autorización y autenticación, a partir del Onboarding el nuevo usuario podrá ingresar al sistema mediante usuario y clave, huella o algún otro método especifique alguno de los anteriores dentro de su arquitectura, también puede recomendar herramientas de industria que realicen estas tareas y robustezca su aplicación.

El sistema utiliza usa base de datos de auditoría que registra todas las acciones del cliente y cuenta con un mecanismo de persistencia de información para clientes frecuentes, para este caso proponga una alternativa basada en patrones de diseño que relacione los componentes que deberían interactuar para conseguir el objetivo.

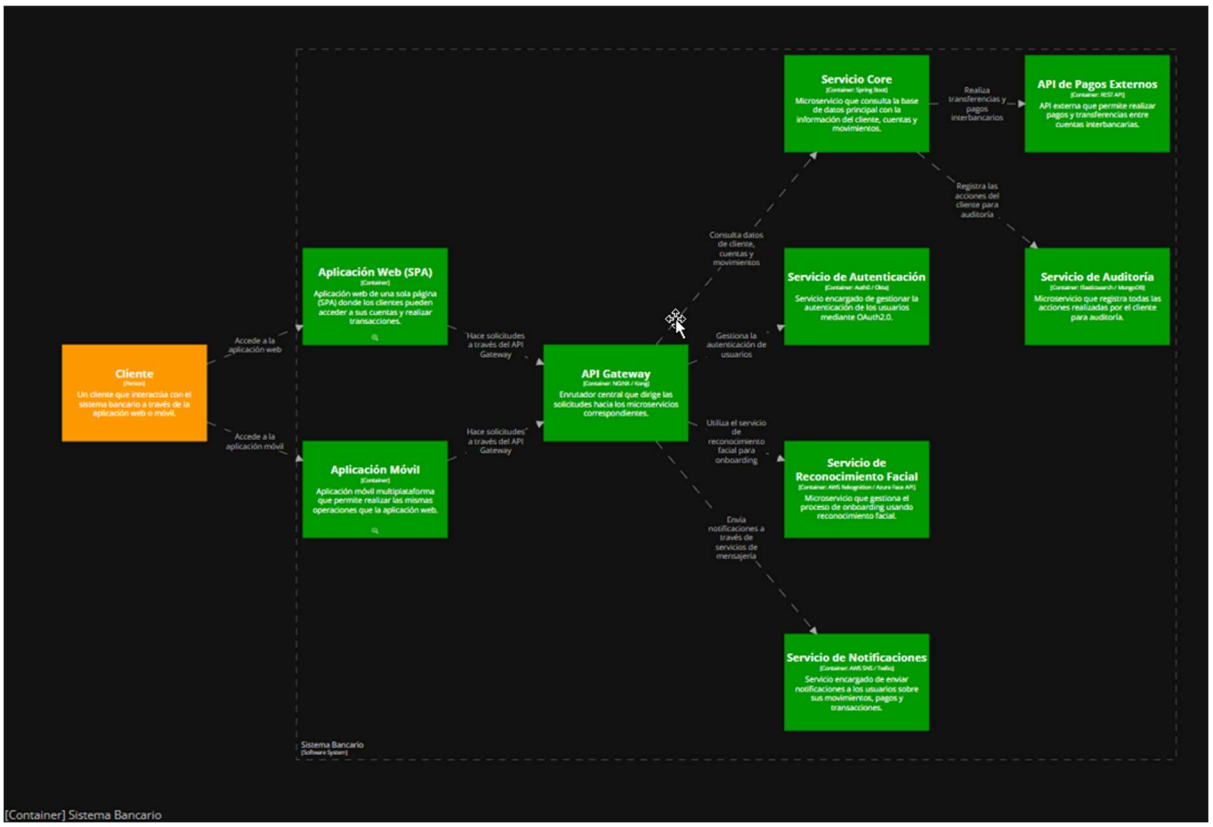
Para obtener los datos del cliente el sistema pasa por una capa de integración compuesta por un api Gateway y consume los servicios necesarios de acuerdo con el tipo de transacción, inicialmente usted cuenta con 3 servicios principales, consulta de datos básicos, consulta de movimientos y transferencias, que realiza llamados a servicios externos dependiendo del tipo, si considera que debería agregar más servicios para mejorar la repuesta de información a sus clientes, es libre de hacerlo.

**Consideraciones**.

a) Para este reto, mencione aquellos elementos normativos que podrían ser importantes a la hora de crear aplicaciones para entidades financieras, Ejemplo ley de datos personales, seguridad etc.
b) Garantice en su arquitectura, alta disponibilidad (**HA**), tolerancia a fallos, recuperación ante desastres (**DR**), Seguridad y Monitoreo, Excelencia operativa y auto-healing.
c) Sí lo considera necesario, su arquitectura puede contener elementos de infraestructura en nube como Azure o AWS, garantice baja latencia en sus servicios, cuenta con presupuesto para esto.
d) En lo posible plantee una arquitectura desacoplada con elementos y reusables y cohesionados para otros componentes que puedan adicionarse en el futuro.

# ARQUITECTURA PROPUESTA

Apoyar a BANCO en el desarrollo e implementación de una arquitectura que permita integrar los sistemas del core bancario llevando a un esquema Nube.


Aplicación Web
[Component: Angular, Tailwind CSS, ReactJS]
Frontend para clientes basado en navegadores.
Aplicación Web (SPA)
[Container]
[Component] Sistema Bancario - Aplicación Web (SPA)

# EJERCICIO PRÁCTICO DE ARQUITECTURA

## 1. Visión General

El sistema de banca por internet de BP permitirá a los usuarios consultar su histórico de movimientos, realizar transferencias y pagos entre cuentas propias e interbancarias. La información de los clientes provendrá de dos sistemas: una plataforma. Además, el sistema contará con mecanismos de autenticación Auth0, Reconocimiento facial, notificación y auditoría.

## 2. Componentes Principales

**a) Frontend**

- **SPA (Single Page Application)**: Se recomienda el uso de **Angular** o **React** debido a su capacidad de manejar interfaces dinámicas, modularidad y reutilización de componentes.
- **Aplicación Móvil**: Se recomienda **Flutter** o **React Native** ya que permiten desarrollo multiplataforma con una sola base de código, reduciendo costos y tiempos de desarrollo, además de tener soporte sólido para integración con autenticación biométrica.

**b) Backend**

- **API Gateway**: Uso de **AWS API Gateway**, **Azure API Management** o **Kong Gateway** para la gestión centralizada del tráfico y autenticación.
- **Microservicios**:
    - **Servicio de Autenticación y Autorización (OAuth 2.0)**: Se recomienda **Auth0**, **Azure AD B2C** o **Okta** por su facilidad de configuración y seguridad avanzada.
    - **Servicio de Consulta de Datos Básicos del Cliente**: Conectará con la plataforma Core para obtener información esencial.
    - **Servicio de Consulta de Movimientos**: Accederá a los datos históricos del usuario.
    - **Servicio de Transferencias y Pagos**: Gestionará transacciones entre cuentas.
    - **Servicio de Notificaciones**: Integración con **Amazon SNS**, **Twilio** o **Firebase Cloud Messaging (FCM)** para asegurar redundancia en la entrega de mensajes.

- **Servicio de Auditoría**: Uso de **Elasticsearch**, **Splunk** o **Azure Monitor** para almacenamiento y consulta rápida de registros de actividad.
- **Servicio de Onboarding con Reconocimiento Facial**: Uso de **Amazon Rekognition**, **Azure Face API** o **Face++** para autenticación biométrica.
- **Servicio de Persistencia de Información para Clientes Frecuentes**: Uso de **Redis**, **Amazon DynamoDB** o **Hazelcast** para cache de alta disponibilidad.

## 3. Autenticación y Autorización

- **OAuth 2.0 con OpenID Connect**
- Flujo recomendado: **Authorization Code Flow con PKCE** para garantizar máxima seguridad, evitando exposición de credenciales en el cliente.
- Onboarding con reconocimiento facial utilizando **Amazon Rekognition**, **Azure Face API** o **Face++**, con integración a un sistema de identidad gestionado.
- Métodos de acceso adicionales: **Huella digital, Face ID y autenticación multifactor (MFA)** con **Microsoft Authenticator, Google Authenticator o YubiKey**.

## 4. Persistencia de Información para Clientes Frecuentes

Se aplicará el **patrón Cache-aside**, donde los datos de clientes frecuentes se almacenarán en **Redis**, **Amazon DynamoDB** o **Hazelcast** o **herramientas Azure** mejorando tiempos de respuesta y reduciendo carga en la base de datos principal.

## 5. Notificaciones

- **Amazon SNS**: Garantiza alta disponibilidad y escalabilidad para notificaciones push.
- **Twilio**: Para envío de SMS y WhatsApp con mayor confiabilidad en mercados globales.
- **Firebase Cloud Messaging (FCM)**: Ideal para notificaciones en dispositivos móviles.

## 6. Infraestructura y Alta Disponibilidad

- **Despliegue en la nube**: Se recomienda **AWS**, **Azure** o **Google Cloud** por su capacidad de escalabilidad global y cumplimiento de normativas.
- **Balanceadores de carga**: **AWS ALB/NLB**, **Azure Load Balancer** o **NGINX** para distribución del tráfico.

- **Base de datos distribuida**: Uso de **Amazon RDS Multi-AZ**, **Azure SQL Managed Instance** o **Google Cloud Spanner** para alta disponibilidad.
- **Microservicios desacoplados** mediante colas de mensajes **RabbitMQ**, **Amazon SQS**, **Azure Service Bus** o **Apache Kafka**.

## 7. Seguridad y Normativas

- **Cifrado de datos en tránsito y en reposo** usando **TLS 1.3, AES-256**.
- **Cumplimiento de normativas**:
    - **ISO 27001**: Gestión de Seguridad de la Información.
    - **PCI DSS**: Protección de datos de tarjetas.
    - **GDPR y Ley de Protección de Datos Personales**: Protección de información personal.
    - **SOX**: Auditoría de transacciones financieras.
    - **OWASP Top 10**: Protección contra vulnerabilidades web.

## 8. Monitoreo y Recuperación ante Desastres

- **Logs centralizados** con **ELK (Elasticsearch, Logstash, Kibana)**, **Splunk** o **Datadog** para trazabilidad.
- **Alertas y monitoreo** con **Prometheus, Grafana o New Relic**.
- **Backups** automáticos y replicación en múltiples regiones con **AWS Backup, Azure Backup o Google Cloud Backup and DR**.
- **Auto-healing** con **Kubernetes**, **AWS Auto Scaling** o **Azure Scale Sets**.

## 9. Beneficios de la Arquitectura Propuesta

- **Escalabilidad**: Microservicios desacoplados permiten crecimiento modular.
- **Seguridad**: Autenticación robusta y cifrado.
- **Alta disponibilidad**: Balanceadores de carga, bases de datos distribuidas y monitoreo.
- **Baja latencia**: Uso de caché y servicios optimizados en la nube.

Esta arquitectura garantiza una solución segura, escalable y eficiente para la banca por internet de BP.