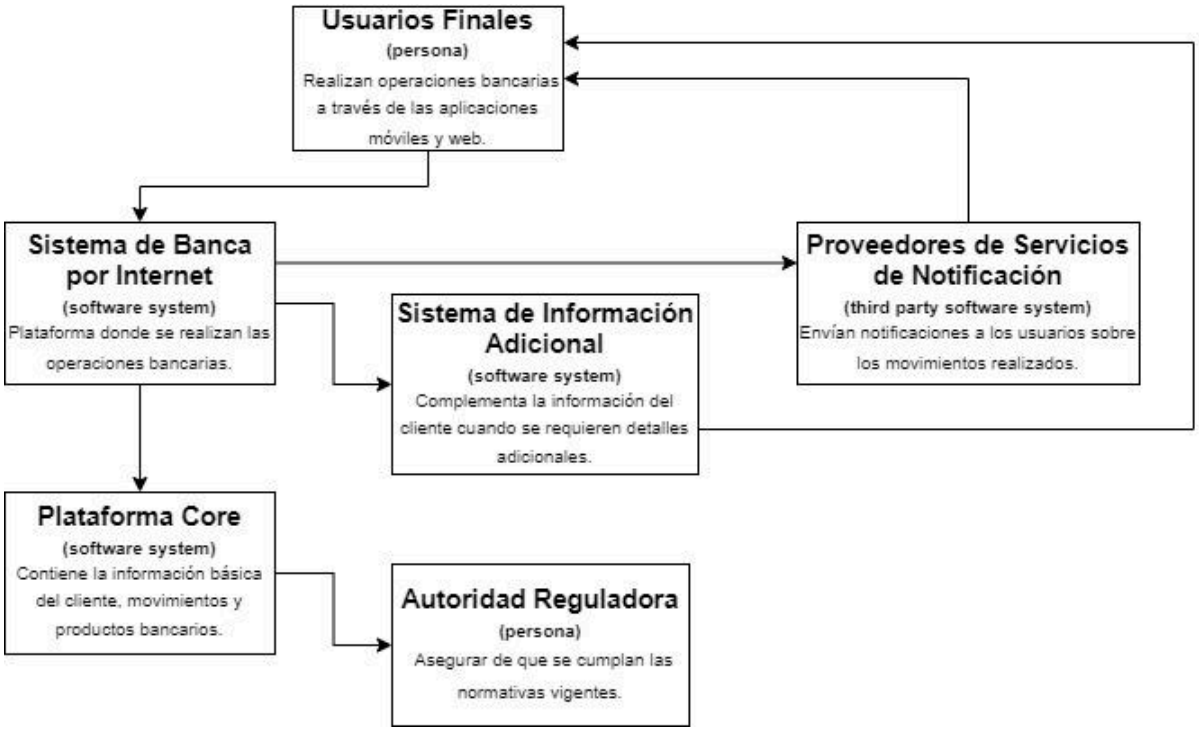
Johnny Paredes

Solución Propuesta para el Sistema de Banca por Internet de BP

1. Modelo C4 (Contexto, Contenedores, Componentes)

1.1 Diagrama de Contexto

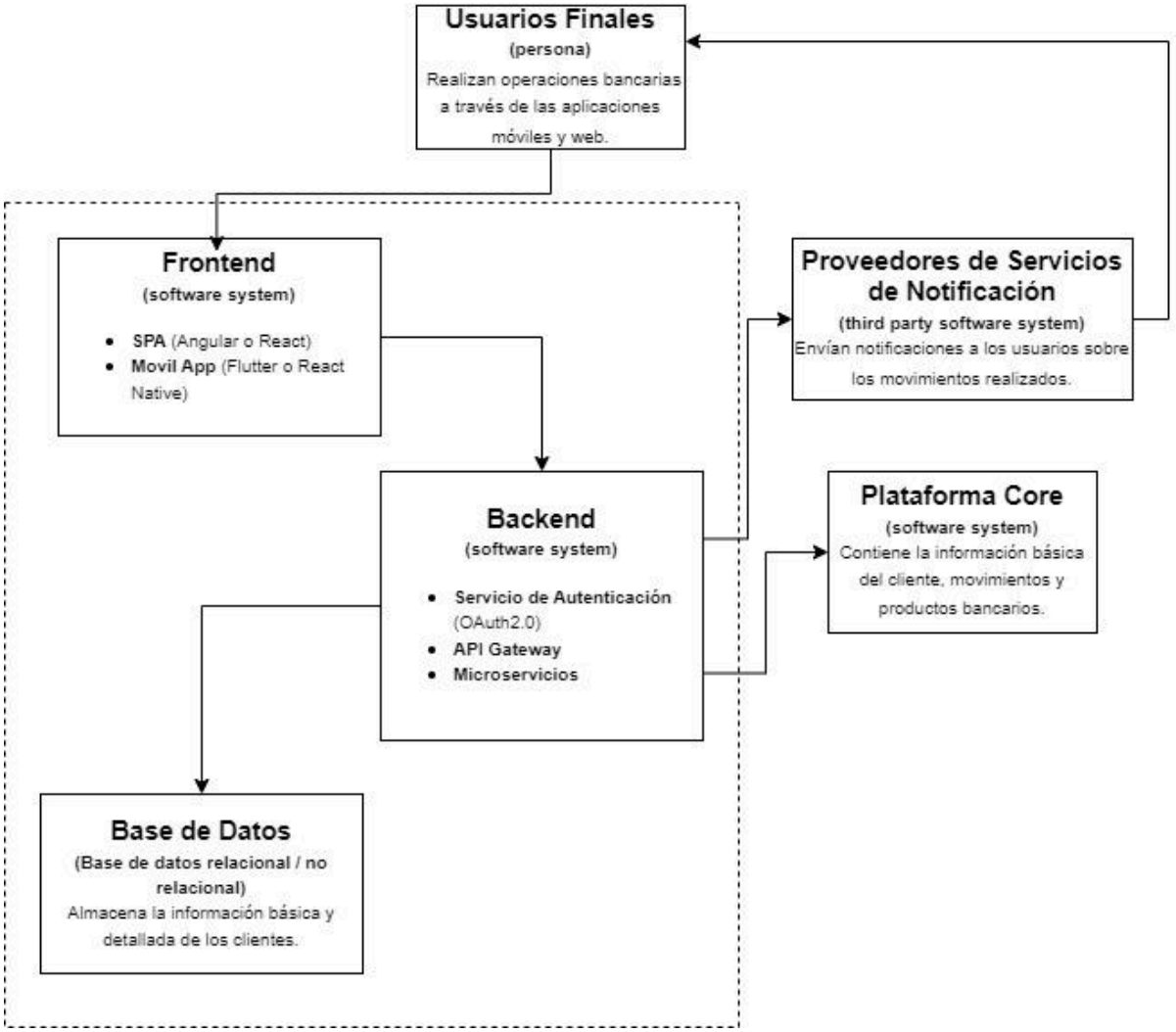
Este diagrama muestra una vista general de los actores involucrados y sus interacciones con el sistema de banca por internet.



- **Usuarios Finales (Clientes):** Realizan operaciones bancarias como transferencias, pagos y consultas de movimientos a través de las aplicaciones móviles y web.
- **Sistema de Banca por Internet (Sistema Principal):** Plataforma donde se realizan las operaciones bancarias.
- **Plataforma Core:** Contiene la información básica del cliente, movimientos y productos bancarios.
- **Sistema de Información Adicional:** Complementa la información del cliente cuando se requieren detalles adicionales.
- **Proveedores de Servicios de Notificación:** Servicios externos que envían notificaciones a los usuarios sobre los movimientos realizados.
- **Autoridad Reguladora:** Se asegura de que se cumplan las normativas vigentes (como la ley de protección de datos personales).

1.2 Diagrama de Contenedores

Este diagrama detalla la arquitectura del sistema, mostrando cómo los diferentes contenedores (aplicaciones, bases de datos, etc.) interactúan entre sí.

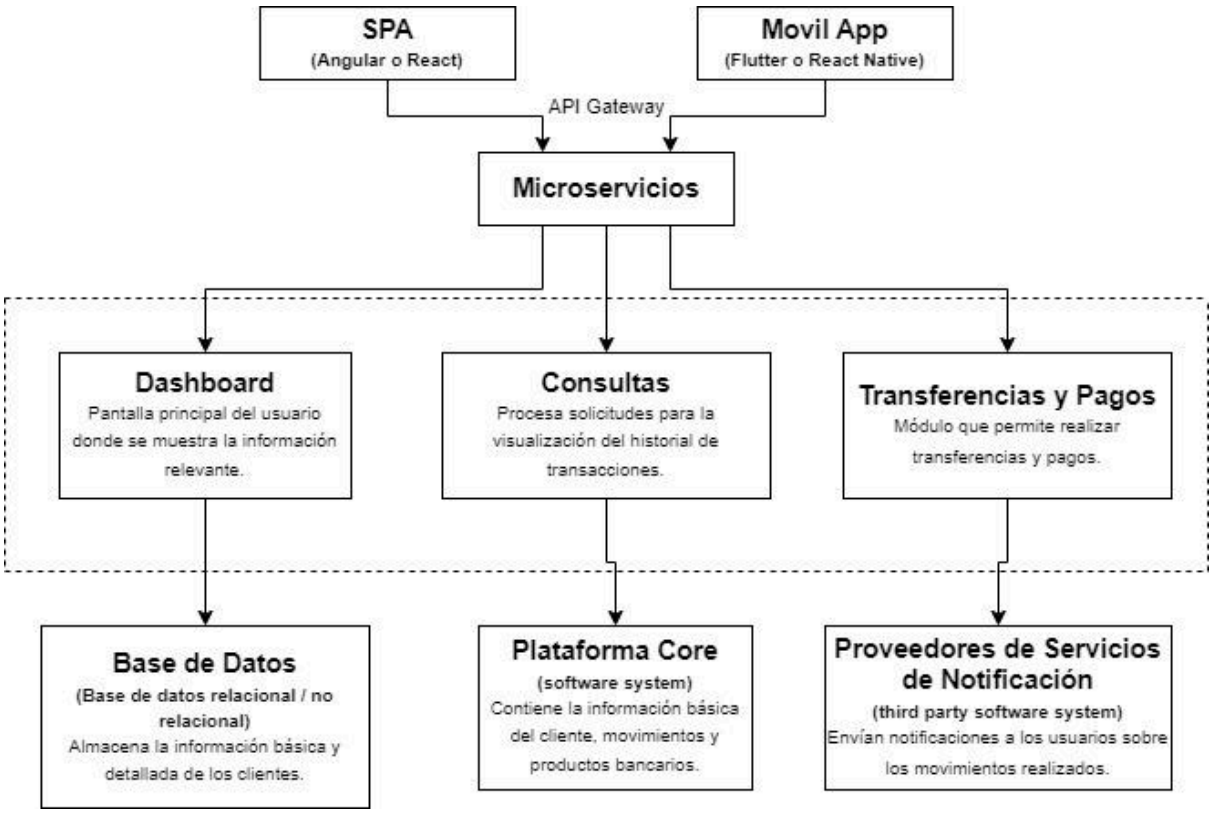


- **Frontend:**
  - **Single Page Application (SPA):** Aplicación web desarrollada con Angular o React para brindar una experiencia fluida al usuario.
  - **Aplicación Móvil:** Desarrollada con Flutter o React Native, permitiendo la compatibilidad multiplataforma (iOS y Android).
- **Backend:**
  - **Servicio de Autenticación:** Implementado con OAuth2.0 para gestionar la autenticación y autorización.
  - **API Gateway:** Punto de entrada único para todas las solicitudes de los usuarios, que redirige las solicitudes a los microservicios correspondientes.
  - **Microservicios:**
    - **Consulta de Datos Básicos:** Interactúa con la Plataforma Core para recuperar la información básica del cliente.
    - **Consulta de Movimientos:** Recupera el historial de movimientos del cliente.
    - **Transferencias:** Maneja las transferencias entre cuentas propias e interbancarias.
    - **Notificaciones:** Envía notificaciones de movimientos a través de los proveedores de servicios de notificación.

- **Bases de Datos:**
    - **Base de Datos de Clientes:** Almacena la información básica y detallada de los clientes.
    - **Base de Datos de Auditoría:** Registra todas las acciones de los clientes para cumplir con las normativas.
    - **Mecanismo de Persistencia para Clientes Frecuentes:** Implementa un patrón de diseño como CQRS (Command Query Responsibility Segregation) para mejorar el rendimiento y la eficiencia del acceso a los datos.

1.3 Diagrama de Componentes

En este diagrama, se desglosan los componentes clave dentro de cada contenedor, detallando sus responsabilidades específicas y cómo se interrelacionan.



- **SPA (Single Page Application):**
    - **Autenticación:** Componente encargado de manejar el login y la gestión de tokens.
    - **Dashboard:** Pantalla principal del usuario donde se muestra la información relevante.
    - **Historial de Movimientos:** Componente que permite la consulta y visualización del historial de transacciones.
    - **Transferencias y Pagos:** Módulo que permite realizar transferencias y pagos.
- **Aplicación Móvil:**
    - **Autenticación:** Maneja la autenticación, ya sea mediante usuario/contraseña, huella o reconocimiento facial.

    - **Onboarding Facial:** Proceso de registro utilizando reconocimiento facial.
    - **Historial de Movimientos:** Visualización del historial de transacciones en la aplicación móvil.
    - **Transferencias y Pagos:** Módulo para realizar transferencias y pagos a través de la aplicación.
- **Servicio de Autenticación:**
    - **Gestión de Tokens:** Creación, validación y revocación de tokens de acceso.
    - **Autorización:** Verificación de permisos y control de acceso a los recursos.
- **API Gateway:**
    - **Enrutamiento:** Redirige las solicitudes entrantes a los microservicios correspondientes.
    - **Seguridad:** Verificación de autenticación y autorización de cada solicitud.
- **Microservicios:**
    - **Consulta de Datos Básicos:** Maneja solicitudes de datos básicos desde la Plataforma Core.
    - **Consulta de Movimientos:** Procesa solicitudes para la visualización del historial de transacciones.
    - **Transferencias:** Lógica de transferencia entre cuentas.
    - **Notificaciones:** Envía notificaciones sobre movimientos realizados.
- **Bases de Datos:**
    - **Base de Datos de Clientes:** Gestiona y almacena toda la información del cliente.
    - **Base de Datos de Auditoría:** Registra todas las acciones realizadas por los usuarios para cumplir con normativas.
    - **Persistencia para Clientes Frecuentes:** Implementa caching o CQRS para manejar consultas frecuentes de forma eficiente.

2. Recomendaciones de Autenticación y Autorización

- **OAuth2.0:**
    - **Flujo de Autorización Recomendado:**
        - **Authorization Code Flow:** Para aplicaciones web (SPA) y móviles, permite un alto nivel de seguridad al intercambiar un código de autorización por un token de acceso.
        - **Client Credentials Flow:** Para comunicación entre servicios backend.
- **Autenticación Adicional para Onboarding:**
    - **Reconocimiento Facial:** Integrado con proveedores como Microsoft Azure Face API o AWS Rekognition, proporcionando una capa adicional de seguridad.
    - **Autenticación Multifactora (MFA):** Se recomienda incluir autenticación con huella digital o reconocimiento facial después del onboarding.

3. Normativas y Consideraciones Legales

- **Ley de Protección de Datos Personales:** Asegurar el cumplimiento de normativas como GDPR o la Ley de Protección de Datos Personales en cada jurisdicción donde opere BP.

- **Seguridad de la Información:** Implementar cifrado de datos en reposo y en tránsito (TLS/SSL), así como autenticación y autorización robustas.
- **Normativas Financieras:** Asegurar la conformidad con normativas locales e internacionales de protección al consumidor y prevención de fraudes.

4. Alta Disponibilidad, Tolerancia a Fallos y Recuperación ante Desastres

- **Alta Disponibilidad (HA):** Despliegue de aplicaciones en múltiples zonas de disponibilidad en la nube (AWS o Azure) para asegurar la disponibilidad continua.
- **Tolerancia a Fallos:** Implementar redundancia a nivel de servicios y bases de datos, usando patrones como "Circuit Breaker" para manejar fallos transitorios.
- **Recuperación ante Desastres (DR):** Configuración de backups automáticos y planes de recuperación en diferentes regiones geográficas.

5. Infraestructura en la Nube

- **AWS:** Uso de servicios como EC2, RDS, S3 y CloudFront para asegurar baja latencia y alta disponibilidad.
- **Azure:** Alternativamente, se pueden usar servicios como Azure App Service, Azure SQL Database, Azure Blob Storage y Azure Traffic Manager.
- **Monitoreo y Auto-Healing:** Implementar soluciones como AWS CloudWatch o Azure Monitor para monitoreo en tiempo real y respuestas automáticas a fallos.

6. Desacoplamiento y Reutilización

- **Arquitectura Basada en Microservicios:** Asegura que los servicios sean independientes y puedan escalar de forma autónoma.
- **API First:** Cada servicio se diseñará con una interfaz clara que permita su reutilización en futuros proyectos.
- **Uso de Patrones de Diseño:** Implementar patrones como CQRS para la persistencia de clientes frecuentes y Event Sourcing para manejar eventos de forma eficiente.

7. Criterios de Éxito

La solución propuesta será evaluada en función de su capacidad para cumplir con los requisitos funcionales y no funcionales, así como su adherencia a las normativas y buenas prácticas de seguridad y arquitectura. Además, se evaluará su capacidad para ser escalable, segura, de alta disponibilidad y su eficiencia en costos.