INFORME

Diseño del sistema de Banca por Internet para BP

Versión 1.0

Febrero / 2020

Elaborado por: Edison Gordón

Contenido

1. ANTECEDENTES **3**

2. OBJETIVO **3**

3. ALCANCE **3**

4. DESCRIPCIÓN DE LA SOLUCIÓN **3**

Arquitectura de la solución 3
Componentes 4
Componentes del Diagrama de Contenedores 4
Interacciones entre Contenedores 5
Propósito del Diagrama de Contenedores 6
Componentes del Diagrama de Componentes 6
Interacciones entre Componentes 9
Interacciones entre Componentes 10
CRONOGRAMA DE DESPLIEGUE 11
FASE 1: PLANIFICACIÓN Y DISEÑO (4 SEMANAS) 11
FASE 2: DESARROLLO Y PRUEBAS UNITARIAS (8 SEMANAS) 11
FASE 3: PRUEBAS DE INTEGRACIÓN Y PRUEBAS DE RENDIMIENTO (6 SEMANAS) 11
FASE 4: DESPLIEGUE EN ENTORNO DE PREPRODUCCIÓN (4 SEMANAS) 11
FASE 5: DESPLIEGUE EN PRODUCCIÓN (2 SEMANAS) 12
FASE 6: MONITOREO Y MANTENIMIENTO (CONTINUO) 12
E. Flujo de Datos 12
F. Seguridad 13
Interacciones entre Componentes de Seguridad 14
Propósito 15
Componentes Clave 15
4. Consideraciones de Seguridad 15

1. Antecedentes

En el mundo actual, la banca en línea se ha convertido en una necesidad para los usuarios. Los clientes de BP demandan acceso a sus cuentas y servicios bancarios a través de múltiples canales digitales, como aplicaciones móviles y navegadores web. BP necesita modernizar su infraestructura bancaria y ofrecer una experiencia digital de alta calidad para satisfacer las expectativas de sus clientes y mantenerse competitivo en el mercado.

2. Objetivo

El objetivo principal es desarrollar un sistema de banca por internet robusto, seguro y escalable que permita a los clientes de BP realizar una amplia gama de transacciones y consultas bancarias de manera eficiente y conveniente a través de múltiples canales digitales. El sistema debe integrarse con los sistemas core de BP, cumplir con las normativas vigentes y ofrecer una experiencia de usuario excepcional.

3. Alcance

El alcance de este proyecto incluye:

- ✓ **Desarrollo de una SPA (Single Page Application)** para acceso a través de navegadores web.
- ✓ **Desarrollo de aplicaciones móviles** para plataformas iOS y Android.
- ✓ **Implementación de una API Gateway** para gestionar las solicitudes y el acceso a los servicios.
- ✓ **Desarrollo de microservicios** para las diferentes funcionalidades (cuentas, transferencias, pagos, etc.).
- ✓ **Integración con el sistema core bancario** de BP.
- ✓ **Implementación de medidas de seguridad** robustas (autenticación, autorización, protección de datos, etc.).
- ✓ **Desarrollo de un sistema de notificaciones** para mantener a los clientes informados sobre sus transacciones.
- ✓ **Implementación de un sistema de auditoría** para registrar las actividades del sistema.
- ✓ **Pruebas exhaustivas** del sistema para garantizar su calidad y funcionamiento correcto.
- ✓ **Despliegue del sistema** en un entorno de producción seguro y escalable

4. Descripción de la solución

BP requiere un sistema de banca por internet robusto, seguro y escalable que permita a sus clientes realizar diversas transacciones y consultas de manera eficiente. El sistema debe integrarse con los sistemas existentes de BP, cumplir con las normativas vigentes y ofrecer una experiencia de usuario excepcional a través de múltiples canales.

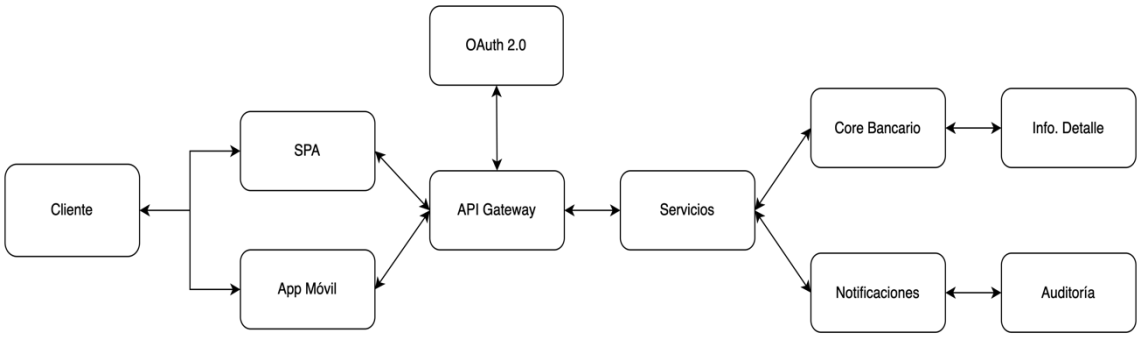
Arquitectura de la solución

A continuación, se presentan los diagramas de contexto, contenedor y componentes, así como otros diagramas relevantes para la arquitectura propuesta.

A. Contexto (C4 Modelo)

Componentes

1. **Sistema:** El sistema de banca por internet de BP se representa como un círculo o rectángulo en el centro del diagrama.
2. **Entidades externas:** Las entidades externas que interactúan con el sistema se representan como rectángulos alrededor del sistema. Estas entidades pueden ser:
    - **Clientes:** Usuarios que acceden al sistema a través de la SPA o la App Móvil.
    - **Core Bancario:** El sistema central de BP donde se encuentran los datos de las cuentas y se realizan las transacciones.
    - **Información Detallada:** Sistemas externos que proporcionan información adicional sobre los clientes o las transacciones.
    - **OAuth 2.0:** El sistema encargado de la autenticación y autorización de los usuarios.
    - **Notificaciones:** El sistema encargado de enviar notificaciones a los clientes (push, SMS, etc.).
    - **Auditoría:** El sistema encargado de registrar las actividades del sistema para fines de auditoría.
3. **Flujos de información:** Las flechas que conectan el sistema con las entidades externas representan los flujos de información entre ellos. Estos flujos pueden ser:
    - Solicitudes de los clientes (consultas, transferencias, pagos, etc.).
    - Respuestas del sistema a los clientes.
    - Datos de las cuentas y transacciones desde el Core Bancario.
    - Información adicional desde sistemas externos.
    - Tokens de autenticación desde OAuth 2.0.
    - Notificaciones a los clientes.
    - Registros de auditoría.



B. Contenedores (C4 Model)

Componentes del Diagrama de Contenedores

1. **SPA (Single Page Application):** Es la aplicación web que los clientes utilizan para acceder al sistema a través de sus navegadores. Está desarrollada con React y proporciona una interfaz de usuario moderna e interactiva.

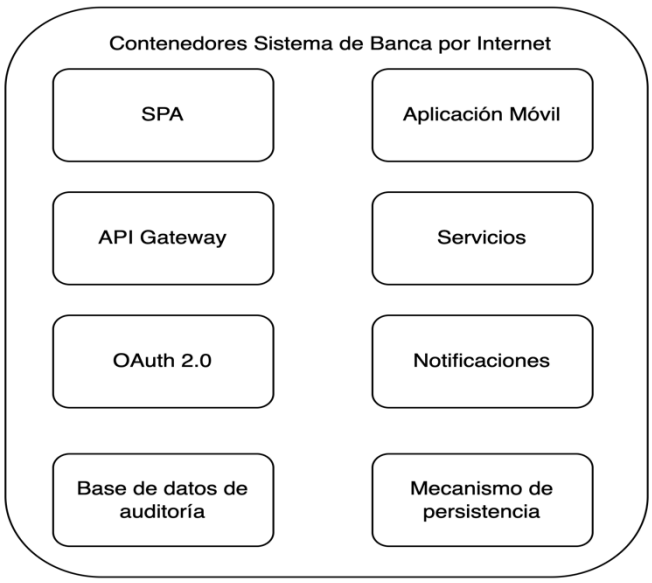
2. **Aplicación Móvil:** Son las aplicaciones nativas para dispositivos iOS y Android que permiten a los clientes acceder al sistema desde sus teléfonos móviles. Pueden estar desarrolladas con React Native o Flutter.
3. **API Gateway:** Es el punto de entrada único para todas las solicitudes que llegan al sistema desde la SPA y la aplicación móvil. Actúa como un proxy inverso, gestiona la autenticación y autorización, el rate limiting y otras funciones de seguridad. Se puede implementar con Kong o Apigee.
4. **Servicios:** Son los microservicios que implementan las diferentes funcionalidades del sistema. Cada servicio se encarga de una tarea específica, como la gestión de cuentas, transferencias, pagos, etc. Están desarrollados con Spring Boot (Java) o Node.js y se comunican entre sí a través de APIs.
    - **Autenticación:** Gestiona la autenticación de los usuarios utilizando Spring Security o Node.js Passport y se integra con el sistema OAuth 2.0.
    - **Cuentas:** Gestiona la información de las cuentas de los clientes.
    - **Transferencias:** Permite a los clientes realizar transferencias entre cuentas.
    - **Pagos:** Permite a los clientes realizar pagos de servicios.
    - **Movimientos:** Consulta y gestiona los movimientos de las cuentas.
    - **Usuarios:** Gestiona la información de los usuarios del sistema.
5. **OAuth 2.0:** Es el sistema encargado de la autenticación y autorización de los usuarios. Se puede implementar con Keycloak o Auth0.
6. **Notificaciones:** Es el sistema encargado de enviar notificaciones a los clientes sobre sus transacciones y otras actividades. Utiliza Firebase Cloud Messaging o OneSignal para notificaciones push y Twilio o Nexmo para notificaciones SMS.
7. **Base de datos de auditoría:** Es la base de datos donde se registran todas las actividades del sistema para fines de auditoría. Se puede implementar con PostgreSQL.
8. **Base de datos transaccional:** Es la base de datos principal del sistema donde se almacena la información de las cuentas y transacciones de los clientes. Se puede implementar con MySQL o PostgreSQL.
9. **Caché:** Es un sistema de almacenamiento en caché que se utiliza para mejorar el rendimiento del sistema. Se puede implementar con Redis.
10. **Bus de mensajes:** Es un sistema de mensajería que se utiliza para la comunicación asíncrona entre los servicios. Se puede implementar con RabbitMQ o Kafka.

Interacciones entre Contenedores

1. La SPA y la aplicación móvil se comunican con la API Gateway para acceder a los servicios del sistema.
2. La API Gateway gestiona las solicitudes y las enruta a los servicios correspondientes.
3. Los servicios se comunican entre sí y con la base de datos transaccional, la base de datos de auditoría, el caché y el bus de mensajes para realizar sus tareas.
4. El sistema OAuth 2.0 se encarga de la autenticación y autorización de los usuarios.
5. El sistema de notificaciones envía notificaciones a los clientes a través de push o SMS.

Propósito del Diagrama de Contenedores

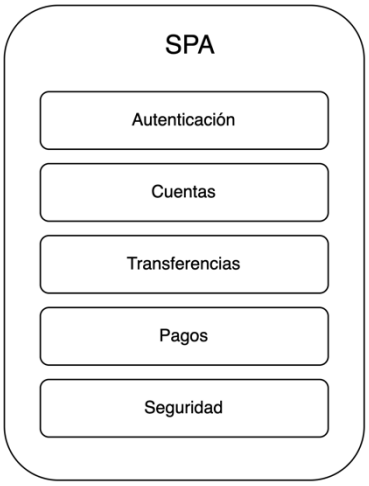
1. **Visualizar la arquitectura del sistema:** El diagrama de contenedores proporciona una visión general de la arquitectura del sistema y cómo sus diferentes componentes interactúan entre sí.
2. **Facilitar la comunicación:** El diagrama de contenedores es una herramienta útil para comunicar la arquitectura del sistema a los diferentes stakeholders del proyecto.
3. **Identificar los puntos de despliegue:** El diagrama de contenedores muestra las unidades de despliegue del sistema, lo que facilita la planificación del despliegue y la gestión de la infraestructura.



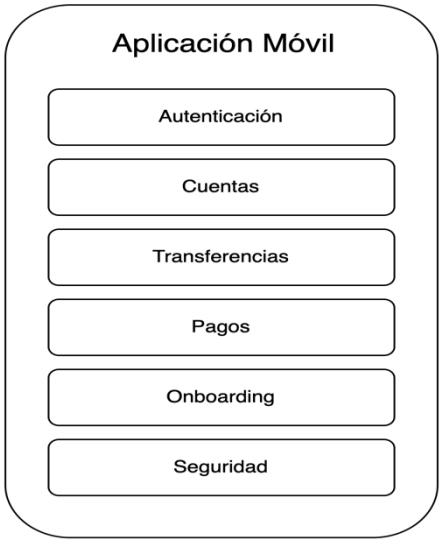
C. Componentes (C4 Model)

Componentes del Diagrama de Componentes

1. **SPA (Single Page Application):** Es la aplicación web que los clientes utilizan para acceder al sistema a través de sus navegadores. Está desarrollada con React y se divide en los siguientes componentes:
    - **Autenticación:** Permite a los usuarios iniciar sesión y gestionar sus credenciales.
    - **Cuentas:** Permite a los usuarios consultar y gestionar sus cuentas bancarias.
    - **Transferencias:** Permite a los usuarios realizar transferencias entre cuentas.
    - **Pagos:** Permite a los usuarios realizar pagos de servicios.
    - **Movimientos:** Permite a los usuarios consultar y gestionar los movimientos de sus cuentas.
    - **Usuarios:** Permite a los usuarios gestionar su información personal.
    - **Seguridad:** Implementa medidas de seguridad en el lado del cliente.

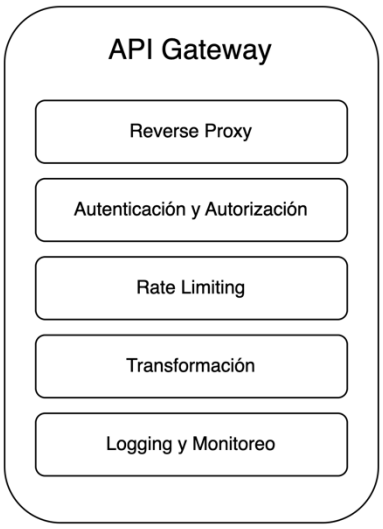


2. **Aplicación Móvil:** Son las aplicaciones nativas para dispositivos iOS y Android que permiten a los clientes acceder al sistema desde sus teléfonos móviles. Pueden estar desarrolladas con React Native o Flutter y se dividen en componentes similares a la SPA, con la adición del componente **Onboarding** para la incorporación de nuevos usuarios.



3. **API Gateway:** Es el punto de entrada único para todas las solicitudes que llegan al sistema desde la SPA y la aplicación móvil. Actúa como un proxy inverso, gestiona la autenticación y autorización, el rate limiting y otras funciones de seguridad. Se divide en los siguientes componentes:
    - **Reverse Proxy:** Actúa como un proxy inverso para proteger los servicios internos.
    - **Autenticación y Autorización:** Gestiona la autenticación y autorización de los usuarios.
    - **Rate Limiting:** Limita la cantidad de solicitudes que pueden llegar al sistema.
    - **Transformación:** Transforma las solicitudes y respuestas entre la SPA/aplicación móvil y los servicios.

- **Logging y Monitoreo:** Registra las actividades del sistema y monitoriza su rendimiento.



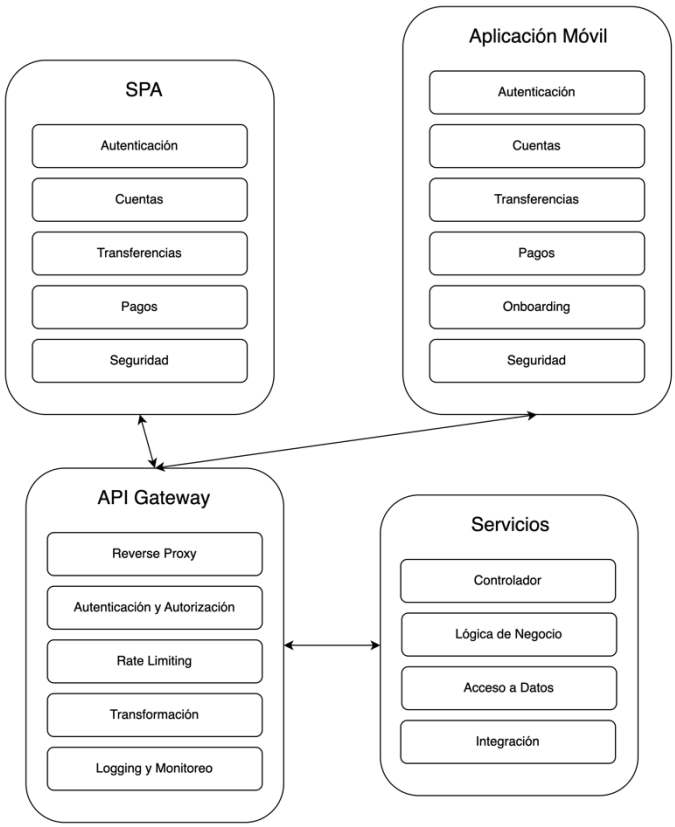
4. **Servicios:** Son los microservicios que implementan las diferentes funcionalidades del sistema. Cada servicio se encarga de una tarea específica y se divide en los siguientes componentes:
   - **Controlador:** Recibe las solicitudes y las dirige a la lógica de negocio.
   - **Lógica de Negocio:** Implementa las reglas de negocio y la lógica de la aplicación.
   - **Acceso a Datos:** Accede a la base de datos y otros sistemas externos para obtener y almacenar datos.
   - **Integración:** Se integra con otros servicios y sistemas externos.

5. **OAuth 2.0:** Es el sistema encargado de la autenticación y autorización de los usuarios.
6. **Notificaciones:** Es el sistema encargado de enviar notificaciones a los clientes sobre sus transacciones y otras actividades.
7. **Base de datos de auditoría:** Es la base de datos donde se registran todas las actividades del sistema para fines de auditoría.
8. **Base de datos transaccional:** Es la base de datos principal del sistema donde se almacena la información de las cuentas y transacciones de los clientes.

9. **Caché:** Es un sistema de almacenamiento en caché que se utiliza para mejorar el rendimiento del sistema.
10. **Bus de mensajes:** Es un sistema de mensajería que se utiliza para la comunicación asíncrona entre los servicios.

Interacciones entre Componentes

- Los componentes de la SPA y la aplicación móvil se comunican con la API Gateway para acceder a los servicios del sistema.
- Los componentes de la API Gateway interactúan con los servicios para gestionar las solicitudes.
- Los componentes de los servicios se comunican entre sí y con la base de datos transaccional, la base de datos de auditoría, el caché y el bus de mensajes para realizar sus tareas.
- El sistema OAuth 2.0 se encarga de la autenticación y autorización de los usuarios.
- El sistema de notificaciones envía notificaciones a los clientes a través de push o SMS.



D. Despliegue

A. **Infraestructura:**
   - **Nube (AWS / Azure):** Se utilizará un proveedor de servicios en la nube para alojar la infraestructura del sistema.
   - **Regiones y Zonas de Disponibilidad:** Se desplegará la infraestructura en múltiples regiones y zonas de disponibilidad para garantizar la alta disponibilidad y tolerancia a fallos.

- **Redes Virtuales (VPC):** Se crearán redes virtuales privadas para aislar y proteger los recursos del sistema.
- **Grupos de Seguridad:** Se configurarán grupos de seguridad para controlar el tráfico de red y restringir el acceso a los recursos.
- **Balanceadores de Carga:** Se utilizarán balanceadores de carga para distribuir el tráfico entre las diferentes instancias de los componentes del sistema.
- **DNS:** Se configurará un servicio DNS para gestionar los nombres de dominio del sistema.
- **CDN:** Se utilizará una CDN para distribuir la SPA y otros recursos estáticos a los usuarios de manera más eficiente.
- **Monitoreo y Alertas:** Se implementará un sistema de monitoreo y alertas para supervisar el rendimiento y la disponibilidad del sistema.

B. **Componentes:**

- **SPA (Single Page Application):** Se desplegará en un servicio de almacenamiento de objetos como S3 (AWS) o Azure Blob Storage y se servirá a través de una CDN.
- **Aplicación Móvil:** Se distribuirá a través de las tiendas de aplicaciones (App Store y Google Play).
- **API Gateway:** Se desplegará en instancias de máquinas virtuales (EC2 en AWS o Azure VM en Azure) o en un clúster de Kubernetes.
- **Servicios:** Se desplegarán en instancias de máquinas virtuales o en un clúster de Kubernetes.
- **OAuth 2.0:** Se desplegará en instancias de máquinas virtuales o en un clúster de Kubernetes.
- **Notificaciones:**
  - **Push:** Se utilizarán servicios de terceros como Firebase Cloud Messaging o OneSignal.
  - **SMS:** Se utilizarán servicios de terceros como Twilio o Nexmo.
- **Base de datos de auditoría:** Se desplegará como un servicio gestionado de base de datos relacional como RDS PostgreSQL (AWS) o Azure Database for PostgreSQL.
- **Base de datos transaccional:** Se desplegará como un servicio gestionado de base de datos relacional como RDS MySQL (AWS) o Azure Database for MySQL.
- **Caché:** Se desplegará como un servicio gestionado de caché como Redis Cloud (AWS) o Azure Cache for Redis.
- **Bus de mensajes:** Se desplegará como un servicio gestionado de mensajería como RabbitMQ Cloud (AWS) o Azure Service Bus.

Interacciones entre Componentes

- La SPA y la aplicación móvil se conectarán a la API Gateway a través de HTTPS.
- La API Gateway se comunicará con los servicios a través de HTTP o HTTPS.
- Los servicios se comunicarán entre sí y con la base de datos transaccional, la base de datos de auditoría, el caché y el bus de mensajes.
- El sistema OAuth 2.0 se utilizará para la autenticación y autorización de los usuarios.
- El sistema de notificaciones se utilizará para enviar notificaciones a los clientes.

Cronograma de Despliegue

Este cronograma se divide en fases clave, cada una con actividades específicas y entregables definidos.

Fase 1: Planificación y Diseño (4 semanas)

- **Actividades:***
  - Definición detallada de los requisitos del sistema.
  - Diseño de la arquitectura de la solución (diagramas de contexto, contenedor, componentes, despliegue, flujo de datos y seguridad).
  - Selección de tecnologías y herramientas.
  - Planificación de la infraestructura en la nube (AWS/Azure).
  - Diseño de la estrategia de pruebas y QA.
  - Elaboración del plan de implementación y cronograma detallado.
- **Entregables:***
  - Documento de requisitos detallados.
  - Diseño de la arquitectura de la solución.
  - Plan de implementación y cronograma.

Fase 2: Desarrollo y Pruebas Unitarias (8 semanas)

- **Actividades:***
  - Desarrollo de los microservicios (Autenticación, Cuentas, Transferencias, etc.).
  - Desarrollo de la API Gateway.
  - Desarrollo de la SPA y la aplicación móvil.
  - Implementación de la seguridad (OAuth 2.0, etc.).
  - Pruebas unitarias de cada componente.
- **Entregables:***
  - Código fuente de los microservicios, API Gateway, SPA y aplicación móvil.
  - Pruebas unitarias y documentación.

Fase 3: Pruebas de Integración y Pruebas de Rendimiento (6 semanas)

- **Actividades:***
  - Integración de los microservicios y componentes.
  - Pruebas de integración para verificar el correcto funcionamiento del sistema en su conjunto.
  - Pruebas de rendimiento para evaluar la capacidad del sistema bajo carga.
  - Pruebas de seguridad para identificar vulnerabilidades.
- **Entregables:***
  - Entorno de pruebas integrado.
  - Informes de pruebas de integración, rendimiento y seguridad.

Fase 4: Despliegue en Entorno de Preproducción (4 semanas)

- **Actividades:***

  - Configuración del entorno de preproducción en la nube.
  - Despliegue del sistema en el entorno de preproducción.
  - Pruebas finales en el entorno de preproducción para simular el entorno real.
  - Ajustes y correcciones necesarias.
- **Entregables:***
  - Entorno de preproducción configurado y funcional.
  - Documentación de despliegue.

Fase 5: Despliegue en Producción (2 semanas)

- **Actividades:***
  - Despliegue del sistema en el entorno de producción.
  - Monitoreo y seguimiento del sistema en producción.
  - Resolución de problemas y ajustes finales.
- **Entregables:***
  - Sistema en producción y funcionando correctamente.

Fase 6: Monitoreo y Mantenimiento (Continuo)

- **Actividades:***
  - Monitoreo continuo del sistema para garantizar su correcto funcionamiento.
  - Mantenimiento y actualizaciones del sistema.
  - Soporte técnico a los usuarios.

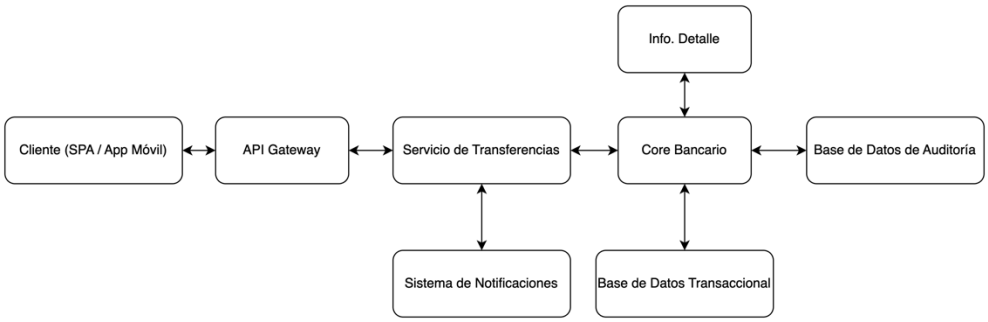
E. Flujo de Datos

- ✓ Cliente: El cliente interactúa con el sistema a través de la SPA (Single Page Application) o la App Móvil, enviando solicitudes para realizar diversas operaciones bancarias (consultas, transferencias, pagos, etc.).
- ✓ SPA / App: La SPA o App Móvil recibe las solicitudes del cliente y las formatea para enviarlas a la API Gateway.
- ✓ API Gateway: La API Gateway actúa como punto de entrada único al sistema, gestionando las solicitudes y enrutándolas a los servicios correspondientes.
- ✓ Servicios: Los servicios son los encargados de procesar las solicitudes y realizar las operaciones bancarias. Interactúan con el Core Bancario para acceder a la información de las cuentas y realizar las transacciones.
- ✓ Core Bancario: El Core Bancario es el sistema central del banco donde se almacenan los datos de las cuentas y se realizan las transacciones.
- ✓ Información Detallada: Los servicios pueden consultar sistemas externos para obtener información adicional sobre los clientes o las transacciones.
- ✓ Autenticación: El sistema de autenticación verifica la identidad del cliente antes de permitir el acceso a los servicios.
- ✓ Notificaciones: El sistema de notificaciones envía mensajes a los clientes para informarles sobre sus transacciones y otras actividades.

- ✓ Auditoría: El sistema de auditoría registra todas las actividades del sistema para fines de seguimiento y cumplimiento normativo.

Interacción

1. Cliente (SPA/App Móvil) inicia transferencia.
2. Solicitud llega al API Gateway.
3. API Gateway autentica y autoriza al cliente.
4. API Gateway enruta la solicitud al servicio de transferencias.
5. Servicio de transferencias consulta datos del cliente (Core Bancario, Info. Detalle).
6. Servicio de transferencias realiza la transferencia (Core Bancario / Servicios externos).
7. Se registra la transacción en la base de datos de auditoría.
8. Se envían notificaciones (Push / SMS) al cliente.
9. Se actualiza la información en la base de datos transaccional y caché.



F. Seguridad

Autenticación:

- **Autenticación Multifactor (MFA):** Se requerirá MFA para todos los usuarios, utilizando métodos como contraseñas, códigos enviados por SMS, autenticación biométrica o tokens de hardware.
- **Gestión de Identidad y Acceso (IAM):** Se utilizará un sistema IAM para gestionar las identidades de los usuarios y controlar su acceso a los recursos del sistema.
- **Single Sign-On (SSO):** Se implementará SSO para permitir a los usuarios acceder a múltiples aplicaciones con una sola autenticación.

Autorización:

- **Control de Acceso Basado en Roles (RBAC):** Se utilizará RBAC para asignar permisos a los usuarios en función de sus roles en la organización.
- **Listas de Control de Acceso (ACL):** Se utilizarán ACL para controlar el acceso a recursos específicos, como archivos o bases de datos.

Protección de Datos:

- **Cifrado en Reposo y en Tránsito:** Todos los datos se cifrarán tanto en reposo (almacenados en bases de datos o discos) como en tránsito (durante la comunicación entre componentes).
- **Tokenización:** Se utilizará la tokenización para proteger datos sensibles, como números de tarjetas de crédito.
- **Enmascaramiento de Datos:** Se utilizará el enmascaramiento de datos para ocultar datos sensibles a usuarios no autorizados.
- **Cumplimiento con PCI DSS y GDPR:** El sistema cumplirá con las normativas PCI DSS y GDPR para garantizar la seguridad de los datos de los clientes.

Seguridad de la Aplicación:

- **Pruebas de Penetración:** Se realizarán pruebas de penetración periódicas para identificar vulnerabilidades en la aplicación.
- **Análisis Estático de Código:** Se utilizarán herramientas de análisis estático de código para identificar problemas de seguridad en el código fuente.
- **Gestión de Vulnerabilidades:** Se implementará un proceso de gestión de vulnerabilidades para identificar, evaluar y corregir vulnerabilidades en la aplicación.
- **Protección contra ataques OWASP Top 10:** Se implementarán medidas de seguridad para proteger la aplicación contra los ataques más comunes identificados por OWASP Top 10.

Seguridad de la Infraestructura:

- **Firewalls:** Se utilizarán firewalls para controlar el tráfico de red y bloquear accesos no autorizados.
- **Sistemas de Detección de Intrusiones (IDS):** Se implementarán IDS para detectar y alertar sobre posibles intrusiones en la red.
- **Seguridad de la Red:** Se implementarán medidas de seguridad para proteger la red, como segmentación de red y VPNs.
- **Endurecimiento de Servidores:** Se realizará un endurecimiento de los servidores para reducir la superficie de ataque y eliminar vulnerabilidades.

Monitoreo y Alertas de Seguridad:

- **SIEM (Security Information and Event Management):** Se utilizará un sistema SIEM para recopilar y analizar logs de seguridad de diferentes fuentes y detectar incidentes de seguridad.
- **Alertas en Tiempo Real:** Se configurarán alertas en tiempo real para notificar al equipo de seguridad sobre eventos de seguridad importantes.
- **Auditoría de Seguridad:** Se realizarán auditorías de seguridad periódicas para evaluar la efectividad de las medidas de seguridad.

Interacciones entre Componentes de Seguridad

- Todos los componentes del sistema interactúan con los componentes de seguridad para garantizar la confidencialidad, integridad y disponibilidad de los datos.

- Los componentes de autenticación y autorización se utilizan para controlar el acceso a los recursos del sistema.
- Los componentes de protección de datos se utilizan para proteger los datos sensibles.
- Los componentes de seguridad de la aplicación y la infraestructura se utilizan para proteger el sistema contra ataques.
- Los componentes de monitoreo y alertas de seguridad se utilizan para detectar y responder a incidentes de seguridad.

Propósito

- **Visualizar las medidas de seguridad:** El diagrama de seguridad muestra las diferentes medidas de seguridad que se implementan en el sistema y cómo interactúan entre sí.
- **Planificar la seguridad:** El diagrama de seguridad ayuda a los equipos de seguridad a planificar la implementación de medidas de seguridad y a evaluar su efectividad.
- **Comunicar la seguridad:** El diagrama de seguridad es una herramienta útil para comunicar las medidas de seguridad a los diferentes stakeholders del proyecto.

G. Componentes Clave

- **SPA:** React, Redux, Material UI
- **Aplicación Móvil:** React Native / Flutter, SDKs nativos
- **API Gateway:** Kong / Apigee, plugins de seguridad y transformación
- **Servicios:** Spring Boot (Java) / Node.js (Express), Spring Security / Passport.js
- **OAuth 2.0:** Keycloak / Auth0, configuración de clientes y scopes
- **Notificaciones:** Firebase Cloud Messaging / OneSignal, Twilio / Nexmo
- **Bases de datos:** PostgreSQL / MySQL, JDBC / ORM
- **Caché:** Redis, Spring Data Redis / ioredis
- **Bus de mensajes:** RabbitMQ / Kafka, Spring AMQP / Kafka client
- **Onboarding:** Onfido / Jumio, SDKs y APIs

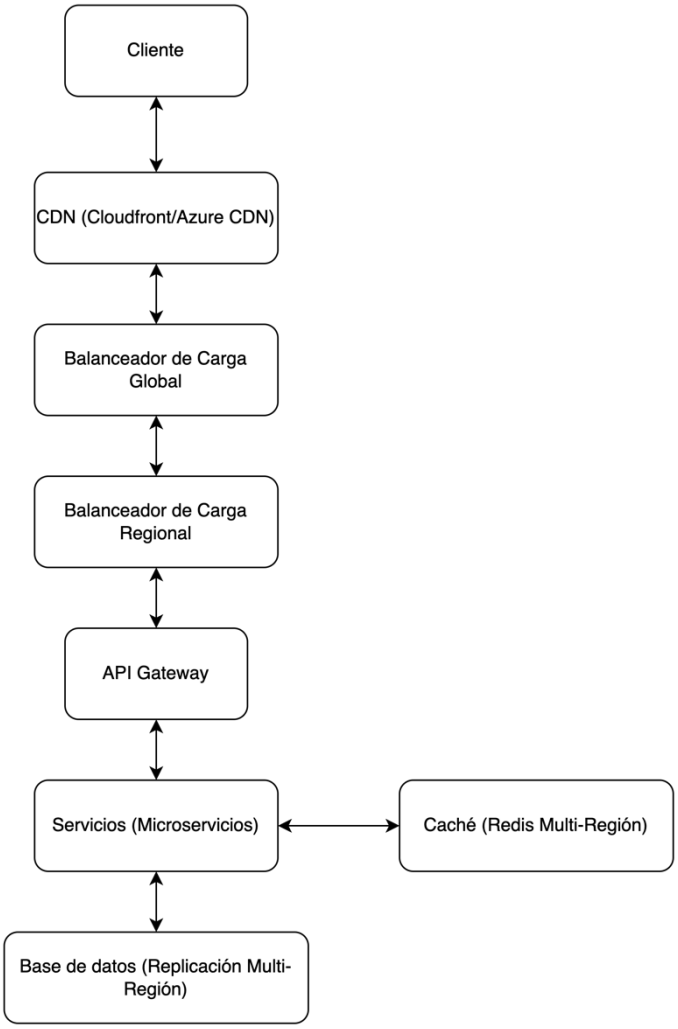
H. Consideraciones de Seguridad

- **Normativa:** Cumplimiento con leyes de protección de datos, regulaciones bancarias, PCI DSS.
- **Autenticación Biométrica:** Integración con SDKs de reconocimiento facial y huella digital.
- **Seguridad en el Desarrollo:** Prácticas de codificación segura, pruebas de seguridad estáticas y dinámicas.
- **Infraestructura Segura:** Configuración robusta de firewalls, VPCs, seguridad de red.
- **Monitoreo y Alertas:** Implementación de herramientas de monitoreo y alertas para detectar anomalías.

I. Implementación de Alta Disponibilidad

Descripción

- **Balanceadores de Carga:** Distribuyen el tráfico entre múltiples instancias de los componentes del sistema para evitar sobrecarga y garantizar la disponibilidad.
- **Replicación de Bases de Datos:** Replican los datos en múltiples zonas de disponibilidad para protegerlos contra fallos y garantizar la disponibilidad.
- **Conmutación por Error Automática:** Permite que el sistema se recupere automáticamente de fallos mediante la conmutación a instancias de respaldo.
- **Escalado Automático:** Permite que el sistema se adapte a cambios en la demanda mediante el escalado automático de los recursos.
- **Infraestructura como Código:** Permite gestionar la infraestructura de manera automatizada y reproducible, lo que facilita la implementación de alta disponibilidad.



Consideraciones

- **Redundancia:** Se deben implementar medidas de redundancia en todos los niveles (hardware, software, datos) para garantizar la alta disponibilidad.
- **Monitoreo:** Se debe monitorear continuamente el sistema para detectar fallos y activar la conmutación por error automática.
- **Pruebas:** Se deben realizar pruebas periódicas para verificar la efectividad de las medidas de alta disponibilidad.

J. Monitoreo

Descripción

- **Recopilación de Logs:** Recopila logs de todos los componentes del sistema en un repositorio centralizado para su análisis.
- **Métricas de Rendimiento:** Monitoriza métricas clave de rendimiento (CPU, memoria, latencia, etc.) para identificar problemas y optimizar el sistema.
- **Alertas y Notificaciones:** Configura alertas para recibir notificaciones sobre eventos críticos o anomalías en el sistema.
- **Paneles de Control:** Crea paneles de control visuales para mostrar el estado y el rendimiento del sistema en tiempo real.

Consideraciones

- **Centralización:** Se recomienda centralizar la recopilación y el análisis de logs y métricas para facilitar la identificación de problemas.
- **Automatización:** Se deben automatizar las tareas de monitoreo y alertas para reducir el esfuerzo manual y mejorar la capacidad de respuesta.
- **Visualización:** Los paneles de control deben ser claros y fáciles de entender para facilitar la toma de decisiones.