

Arquitectura para Sistema de Banca en Línea

Solución propuesta por Javier Fernando carpio Vintimilla

Fecha: 22 de octubre de 2024

Tabla de Contenidos

Contenido

Tabla de Contenidos	2
1. Introducción	4
2. Diagramas de Arquitectura (C4)	5
2.1. Diagrama de Contexto	5
2.2. Diagrama de Contenedores	10
2.3. Diagrama de Componentes	14
Frontend del Sistema de Banca por Internet	14
Módulo Componente de Usuarios - SPA (Single-Page Application)	14
BackEnd del Sistema de Banca por Internet.....	23
Módulo Componente API Gateway (Amazon API Gateway)	23
Módulo Componente Servicio de Autenticación (AWS Cognito).....	27
Módulo Componente Servicio OnBoarding	30
Módulo Componente Servicio Core Bancario	33
Módulo Componente Servicio de Detalles del Cliente	36
Módulo Componente Servicio de Transferencias (ECS/EKS)	39
Módulo Componente Servicio de Notificaciones	41
Módulo Componente Servicio de Auditoría (ECS/EKS).....	45
Módulo Componente Servicio de Auditoría (ECS/EKS).....	47

Módulo Componente Servicio de Cache (Amazon ElastiCache)	49
Base de datos del Sistema de Banca por Internet.....	52
Base de Datos Core (Amazon RDS)	52
Base de Datos Auditoría (Amazon DynamoDB).....	56
Base de Datos de Cache (ElastiCache - Redis).....	59
3. Justificación de Tecnologías y Patrones	62
4. Seguridad y Autenticación.....	62
5. Alta Disponibilidad y Recuperación ante Desastres	63
6. Integración con Servicios Externos y Gestión de Datos	66
7. Consideraciones Normativas	66
8. Manejo de Costos.....	67
9. Conclusiones	69

1. Introducción

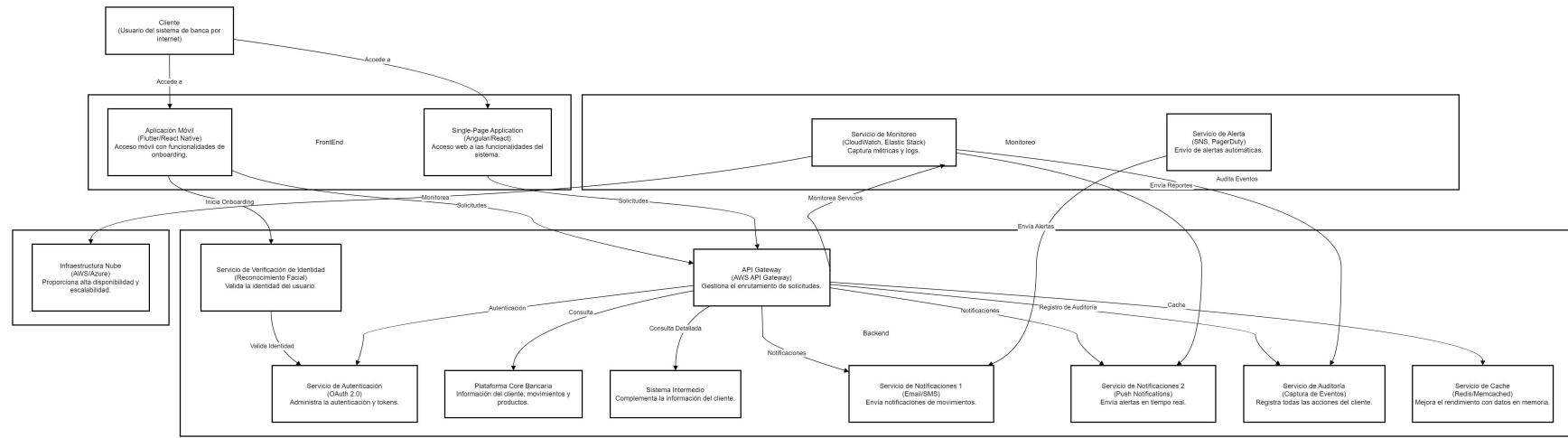
El presente documento describe la arquitectura propuesta para un sistema de banca en línea, encargado de proporcionar servicios financieros a los usuarios, tales como acceso al historial de movimientos, realización de transferencias y pagos entre cuentas. El objetivo es diseñar una solución que garantice un alto nivel de disponibilidad, seguridad y escalabilidad, cumpliendo con los requisitos regulatorios del sector financiero.

La solución está basada en un enfoque de arquitectura de microservicios, utilizando principios de diseño desacoplados y tecnologías modernas para asegurar la interoperabilidad y la facilidad de mantenimiento. Además, se implementarán mecanismos de autenticación robusta, tales como OAuth 2.0, y medidas de alta disponibilidad para proteger los datos del cliente y garantizar la continuidad del servicio.

A lo largo del documento, se presentan diagramas de arquitectura detallados (modelo C4) que ilustran la interacción entre los diferentes componentes, así como la justificación de las tecnologías seleccionadas y las estrategias adoptadas para cumplir con los requisitos de seguridad y normativa bancaria.

2. Diagramas de Arquitectura (C4)

2.1. Diagrama de Contexto



Descripción del Diagrama de Contexto del Sistema de Banca por Internet

El diagrama de contexto muestra la arquitectura general del sistema de banca por internet diseñado para la entidad BP. El sistema está dividido en cuatro secciones principales: **Frontend**, **Backend**, **Infraestructura en la Nube**, y **Monitoreo y Alerta**, las cuales se detallan a continuación:

1. Cliente (Usuario del sistema de banca por internet)

El cliente es el usuario final que interactúa con el sistema para realizar diversas operaciones, como consultar el historial de movimientos, realizar transferencias y pagos. El cliente accede a través de dos canales principales: una aplicación web y una aplicación móvil.

2. Frontend

Single-Page Application (SPA):

Implementada en tecnologías como Angular o React, esta aplicación proporciona acceso web a las funcionalidades del sistema de banca por internet.

Permite al usuario gestionar cuentas, realizar transferencias y consultar movimientos desde un navegador web.

Aplicación Móvil (Flutter/React Native):

Desarrollada con un framework multiplataforma, la aplicación móvil ofrece acceso a las funcionalidades del sistema desde dispositivos móviles.

Incluye capacidades adicionales de **onboarding**, como el registro de nuevos usuarios a través de reconocimiento facial, además de las operaciones esenciales como transferencias y consultas.

3. Backend

API Gateway (Nginx o AWS API Gateway):

Es el punto central de entrada para todas las solicitudes provenientes del frontend.

Gestiona el enrutamiento de las solicitudes a los servicios backend apropiados, asegurando una distribución eficiente del tráfico.

Servicio de Autenticación (OAuth 2.0):

Administra la autenticación de los usuarios utilizando el estándar OAuth 2.0, proporcionando un proceso seguro de inicio de sesión y generación de tokens de acceso.

Gestiona la autenticación inicial para nuevos usuarios durante el proceso de **onboarding**, que incluye la validación de identidad con reconocimiento facial.

Servicio de Verificación de Identidad (Onboarding):

Valida la identidad de los nuevos usuarios mediante el reconocimiento facial y otros métodos de verificación durante el proceso de registro.

Garantiza que solo usuarios verificados puedan completar el registro e iniciar sesión en el sistema.

Plataforma Core Bancaria:

Es el sistema principal que contiene la información básica del cliente, incluyendo datos de cuenta, movimientos y productos.

El API Gateway se comunica con este sistema para recuperar información crítica para las transacciones del usuario.

Sistema Intermedio:

Proporciona datos complementarios del cliente cuando se requiere información más detallada o específica que no está disponible en la plataforma core bancaria.

Servicios de Notificación:

Servicio de Notificaciones 1 (Email/SMS):

Utilizado para enviar notificaciones sobre los movimientos realizados, cumpliendo con las normativas que exigen la notificación al usuario.

Servicio de Notificaciones 2 (Push Notifications):

Envío de alertas en tiempo real a los dispositivos móviles de los usuarios, lo que facilita la comunicación instantánea sobre las transacciones.

Servicio de Auditoría:

Registra todas las acciones del usuario y del sistema, garantizando el cumplimiento normativo y facilitando la auditoría.

Ahora también captura eventos relacionados con el **onboarding** y otras operaciones críticas.

Servicio de Cache (Redis/Memcached):

Almacena en memoria los datos más frecuentes para mejorar el rendimiento del sistema, reduciendo el tiempo de respuesta de las consultas.

4. Infraestructura en la Nube (Azure/AWS)

La infraestructura del sistema está desplegada en una plataforma en la nube (Azure o AWS), proporcionando:

Alta Disponibilidad: Garantiza que los servicios estén siempre accesibles mediante la redundancia y la replicación en múltiples zonas.

Escalabilidad: Ajusta automáticamente la capacidad de los servicios según la demanda del sistema.

Tolerancia a Fallos y Recuperación ante Desastres: Permite la continuidad de los servicios ante fallos mediante mecanismos de recuperación y redundancia.

Baja Latencia: Los servicios en la nube están optimizados para brindar tiempos de respuesta rápidos a los usuarios.

Monitoreo y Auto-Healing: Las herramientas de la nube permiten supervisar el estado del sistema y realizar reparaciones automáticas si se detectan problemas.

5. Monitoreo y Alerta

Servicio de Monitoreo (CloudWatch, Elastic Stack):

Captura métricas y logs de todos los servicios e infraestructura en la nube.

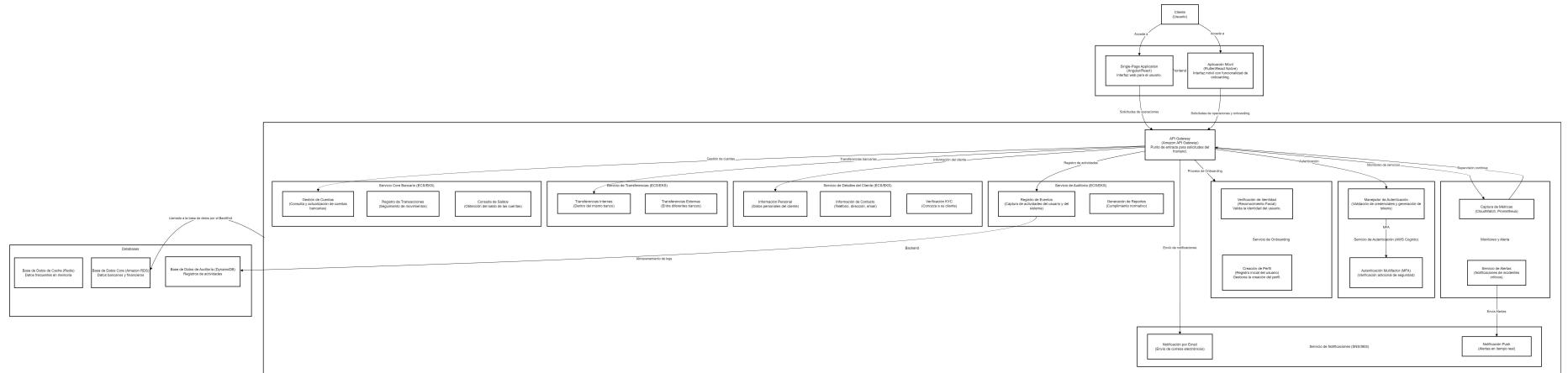
Proporciona una vista en tiempo real del estado del sistema y permite identificar problemas antes de que afecten a los usuarios.

Servicio de Alerta (SNS, PagerDuty):

Envía alertas automáticas en caso de fallos, incidentes o actividades sospechosas.

Puede integrarse con el **Servicio de Notificaciones** para enviar mensajes de alerta a los administradores del sistema.

2.2. Diagrama de Contenedores



Descripción del Diagrama de Contenedores del Sistema de Banca por Internet

El diagrama de contenedores muestra la arquitectura lógica del sistema de banca por internet para la entidad BP, ilustrando cómo los diferentes servicios y aplicaciones interactúan entre sí dentro de la infraestructura de AWS. A continuación, se detalla cada sección y sus componentes:

1. Cliente (Usuario)

El cliente es el usuario final que accede al sistema de banca por internet para realizar operaciones como consultas, transferencias y pagos. Puede interactuar con el sistema a través de dos aplicaciones principales: una aplicación web y una aplicación móvil.

2. Frontend

Single-Page Application (SPA):

Implementada con frameworks como Angular o React, la SPA ofrece una interfaz web dinámica que permite al usuario interactuar con las funciones del sistema desde un navegador.

Todas las solicitudes de datos y acciones son enviadas al backend a través del **API Gateway**.

Aplicación Móvil:

Desarrollada en Flutter o React Native, la aplicación móvil proporciona acceso a la funcionalidad bancaria desde dispositivos móviles, optimizando la experiencia para el entorno móvil.

Incluye el flujo de **onboarding**, permitiendo el registro inicial del usuario mediante **reconocimiento facial** y otros métodos de verificación.

Al igual que la SPA, la aplicación móvil se comunica con el backend a través del **API Gateway** para procesar solicitudes y mostrar la información al usuario.

3. Backend

API Gateway (Amazon API Gateway):

Actúa como el punto central de entrada para todas las solicitudes de las aplicaciones frontend. Dirige las solicitudes a los servicios específicos en el backend según el tipo de operación requerida (autenticación, onboarding, transacciones, notificaciones, etc.).

Servicio de Autenticación (AWS Cognito):

Gestiona la autenticación de usuarios y la emisión de tokens de acceso, garantizando la seguridad del sistema.

Integra capacidades avanzadas como la **autenticación multifactor (MFA)** y el inicio de sesión con redes sociales.

Servicio de Onboarding:

Maneja el flujo de registro inicial de los usuarios, incluyendo la **verificación de identidad** mediante reconocimiento facial y la **creación de perfil** del usuario.

Asegura que solo los usuarios verificados puedan completar el registro e iniciar sesión en el sistema.

Servicio Core Bancario (ECS o EKS):

Es responsable de manejar las operaciones relacionadas con la información bancaria principal, como consultas de saldos y movimientos.

Utiliza servicios de contenedores en AWS (Amazon ECS o EKS) para asegurar una alta disponibilidad y escalabilidad.

Servicio de Detalles del Cliente (ECS o EKS):

Proporciona información adicional y específica del cliente que puede no estar almacenada en la base de datos principal.

Se asegura de complementar la información básica para operaciones detalladas.

Servicio de Transferencias (ECS o EKS):

Gestiona las operaciones de transferencia de fondos entre cuentas del mismo banco o interbancarias.

Asegura la consistencia y trazabilidad de las transacciones.

Servicio de Notificaciones (SNS/SES):

Envía notificaciones a los usuarios sobre las operaciones realizadas. Puede utilizar Amazon SNS para notificaciones push y Amazon SES para notificaciones por correo electrónico.

Este servicio es crucial para cumplir con las normativas de notificación de movimientos bancarios y ahora también envía alertas en caso de incidentes detectados por el sistema de monitoreo.

Servicio de Auditoría (ECS o EKS):

Registra todas las acciones del usuario y las operaciones del sistema, almacenando los registros en una base de datos de auditoría para fines de cumplimiento.

Ha sido mejorado para capturar un rango más amplio de eventos, incluyendo el proceso de **onboarding** y la detección de actividades sospechosas.

Servicio de Cache (Amazon ElastiCache):

Utiliza servicios de cache en memoria (Redis o Memcached) para acelerar el acceso a datos frecuentemente solicitados, mejorando el rendimiento general del sistema.

4. Bases de Datos

Base de Datos Core (Amazon RDS):

Es la base de datos principal del sistema, que almacena toda la información bancaria crítica, como cuentas, transacciones y clientes.

Garantiza la integridad de los datos a través de un sistema de base de datos relacional.

Base de Datos de Auditoría (Amazon DynamoDB):

Utilizada para almacenar registros detallados de auditoría, lo que facilita el seguimiento y la trazabilidad de todas las operaciones realizadas.

Su naturaleza NoSQL permite una gestión eficiente de grandes volúmenes de datos.

Base de Datos de Cache (ElastiCache - Redis):

Optimiza la velocidad de respuesta del sistema al almacenar datos en memoria que se consultan con frecuencia.

Aumenta el rendimiento del sistema reduciendo la carga sobre las bases de datos principales.

5. Infraestructura en AWS

- Todos los componentes del sistema están desplegados en la nube de AWS, lo que proporciona características críticas como:
 - **Alta Disponibilidad:** Los servicios están diseñados para estar siempre disponibles, incluso en caso de fallos.
 - **Escalabilidad Automática:** La infraestructura puede aumentar o reducir su capacidad en función de la demanda.
 - **Recuperación ante Desastres:** Los datos y servicios están replicados para una rápida recuperación.
 - **Monitoreo y Auto-Healing:** Los servicios se supervisan constantemente mediante herramientas como CloudWatch y Prometheus, y las fallas se reparan automáticamente cuando se detectan problemas.

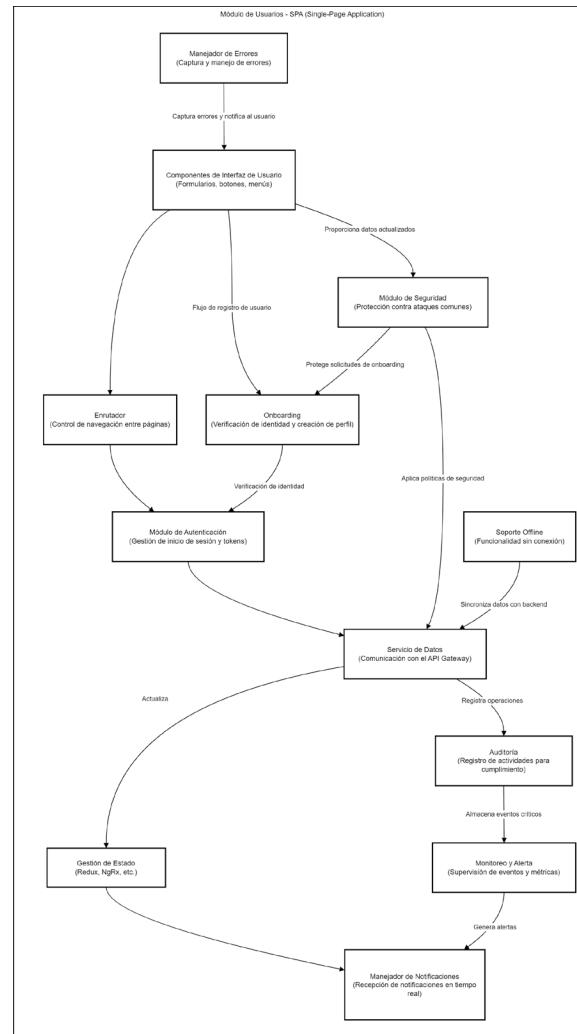
2.3. Diagrama de Componentes

Frontend del Sistema de Banca por Internet

Módulo Componente de Usuarios - SPA (Single-Page Application)

El Módulo de Usuarios - SPA forma parte del frontend del sistema de banca por internet, proporcionando una interfaz interactiva y dinámica para los usuarios que acceden desde navegadores web. Este módulo está diseñado para gestionar la autenticación,

comunicación con los servicios backend, navegación entre vistas, y manejo de notificaciones en tiempo real. A continuación, se detalla la estructura de los componentes y sus interacciones.



Detalle de los Componentes del Módulo de Usuarios - SPA (Single-Page Application)

1. Componentes de Interfaz de Usuario (UIComponents)

- Los componentes de interfaz de usuario son los elementos visuales con los que interactúan los usuarios, incluyendo formularios, botones y menús. Estos componentes permiten a los usuarios realizar tareas como iniciar sesión, consultar su saldo y realizar transferencias. Para la implementación, se recomienda el uso de librerías de componentes como Angular Material o React Bootstrap, que aseguran consistencia en el diseño de la interfaz.

2. Enrutador (Routing)

- El enrutador se encarga de la navegación dentro de la aplicación, controlando el flujo entre las diferentes páginas o vistas. Este componente garantiza que las rutas protegidas sean accesibles solo para usuarios autenticados, utilizando mecanismos de protección como Route Guards.

3. Módulo de Autenticación (AuthModule)

- El módulo de autenticación gestiona el ciclo de vida de la autenticación del usuario, manejando el inicio de sesión, la renovación de tokens y el cierre de sesión. La autenticación se realiza utilizando el estándar OAuth 2.0, y los tokens se almacenan de manera segura en el almacenamiento del navegador.

4. Servicio de Datos (DataService)

- El servicio de datos actúa como intermediario entre la SPA y el backend, enviando solicitudes al API Gateway y gestionando la respuesta de los servicios. Este componente también se encarga de gestionar errores en las solicitudes y de aplicar mecanismos de caché para mejorar el rendimiento.

5. Gestión de Estado (StateManagement)

- La gestión de estado centraliza los datos de la aplicación, manteniendo la información sincronizada en todos los componentes. Esto permite que la interfaz se actualice automáticamente cuando hay cambios en el estado, como la llegada de nuevas notificaciones o la actualización de los datos del usuario.

6. Manejador de Notificaciones (NotificationHandler)

- El manejador de notificaciones se encarga de recibir alertas y notificaciones push en tiempo real, mostrando mensajes relevantes al usuario. Esto se puede lograr utilizando tecnologías como WebSockets o Server-Sent Events (SSE) para la comunicación en tiempo real.

7. Manejador de Errores (ErrorHandler)

- Este componente captura errores que ocurren durante la ejecución de la aplicación, los registra para su análisis y muestra mensajes comprensibles al usuario. Es esencial para mejorar la experiencia de usuario y para la depuración del sistema.

8. Soporte Offline (OfflineSupport)

- El soporte offline proporciona capacidades de acceso a la aplicación sin conexión a internet, permitiendo que los usuarios realicen operaciones básicas y sincronizando los datos cuando la conexión se restablece. La implementación se basa en la arquitectura de Progressive Web Apps (PWA) utilizando Service Workers.

9. Módulo de Seguridad (SecurityModule)

- El módulo de seguridad implementa medidas para proteger la aplicación contra ataques comunes, como Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF). Además, asegura que las credenciales y tokens sean gestionados de manera segura.

10. Monitoreo y Alerta (Monitoring)

- El módulo de monitoreo recopila métricas y supervisa eventos críticos en la aplicación, utilizando herramientas como CloudWatch o Prometheus para la observación. Puede generar alertas automáticas que se envían al manejador de notificaciones en caso de anomalías o problemas detectados. Esto garantiza una rápida respuesta a incidentes y ayuda a mantener la alta disponibilidad del sistema.

11. Onboarding

- El módulo de onboarding permite el registro de nuevos usuarios, gestionando el flujo de verificación de identidad (por ejemplo, mediante reconocimiento facial) y la creación de perfiles. Está integrado con el módulo de autenticación para validar usuarios nuevos y evitar accesos no autorizados.

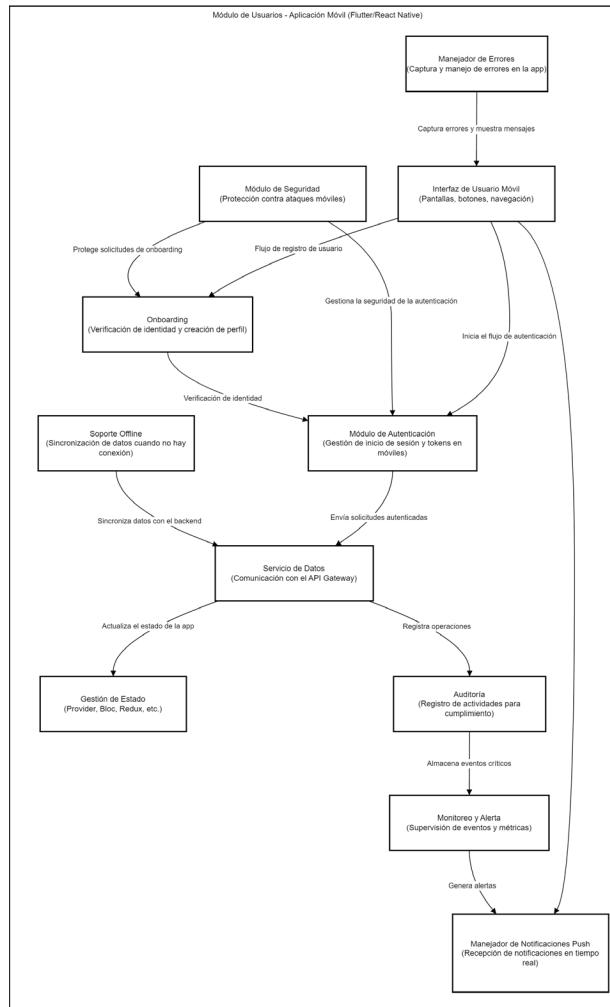
12. Auditoría (Audit)

- El módulo de auditoría registra actividades críticas dentro de la aplicación para asegurar el cumplimiento normativo y proporcionar trazabilidad. Captura eventos como el inicio de sesión, cambios en la configuración de seguridad y errores significativos. Los registros de auditoría se almacenan en una base de datos para futuras revisiones y análisis de cumplimiento.

Consideraciones de Seguridad y Buenas Prácticas

- Aplicar políticas de **Content Security Policy (CSP)** para evitar ataques de inyección de scripts.
- Implementar almacenamiento seguro para los tokens utilizando librerías de seguridad específicas del navegador.
- Configurar un servicio de monitoreo de errores para capturar y registrar errores críticos en la aplicación.
- Asegurar que las operaciones de **onboarding** cumplan con los estándares de verificación de identidad y protección de datos.
- Realizar auditorías periódicas de los registros para detectar comportamientos inusuales o potenciales brechas de seguridad.

Módulo Componente de Usuarios - Aplicación Móvil (Flutter/React Native)



Detalle de los Componentes del Módulo de Usuarios - Aplicación Móvil (Flutter/React Native)

1. Interfaz de Usuario Móvil (UIComponents)

La interfaz de usuario móvil consta de pantallas, botones y elementos de navegación que permiten a los usuarios interactuar con la aplicación. Facilita tareas como el inicio de sesión, la verificación de identidad, la consulta de saldos y la gestión de transferencias. Se recomienda utilizar marcos como Flutter Widgets o React Native Components para crear una interfaz consistente y responsiva.

2. Módulo de Seguridad (SecurityModule)

El módulo de seguridad protege la comunicación de datos y las interacciones dentro de la aplicación, asegurando que se implementen medidas de seguridad como la encriptación de datos y la protección contra ataques móviles comunes. Además, verifica que las solicitudes de autenticación y onboarding sean seguras.

3. Módulo de Autenticación (AuthModule)

Gestiona el ciclo de vida de la autenticación del usuario, incluyendo el inicio de sesión, la gestión de tokens y el cierre de sesión. Utiliza el estándar OAuth 2.0 para asegurar la autenticación y proporciona soporte para autenticación multifactor (MFA) en dispositivos móviles.

4. Soporte Offline (OfflineSupport)

Este módulo permite que la aplicación funcione sin conexión a internet, proporcionando capacidades de sincronización de datos. Cuando el dispositivo se reconecta, sincroniza los datos con el backend, garantizando que la información esté actualizada. Implementado mediante arquitecturas como Service Workers o Cache Storage.

5. Servicio de Datos (DataService)

Actúa como intermediario entre la aplicación y el backend, enviando solicitudes al API Gateway y gestionando las respuestas. El servicio de datos es responsable de aplicar políticas de caché para optimizar el rendimiento y de manejar errores que ocurran durante la comunicación con los servicios de backend.

6. Gestión de Estado (StateManagement)

Centraliza el estado de la aplicación, asegurando que la información esté sincronizada en todos los componentes. Facilita la actualización automática de la interfaz cuando hay cambios en el estado de la aplicación. Los frameworks comunes para esta funcionalidad incluyen Provider, Bloc y Redux.

7. Manejador de Notificaciones Push (NotificationHandler)

Recibe y gestiona notificaciones push en tiempo real, presentando alertas relevantes para el usuario, como notificaciones de transferencias o alertas de seguridad. La comunicación en tiempo real se puede implementar con tecnologías como Firebase Cloud Messaging (FCM).

8. Manejador de Errores (ErrorHandler)

Captura errores que ocurren durante la ejecución de la aplicación, los registra para su posterior análisis y muestra mensajes comprensibles al usuario. Mejora la experiencia del usuario y facilita la depuración del sistema.

9. Monitoreo y Alerta (Monitoring)

El módulo de monitoreo recopila métricas y supervisa la actividad de la aplicación en tiempo real. Utiliza herramientas como Firebase Performance Monitoring o Sentry para observar el rendimiento y detectar incidentes. Genera alertas automáticas en caso de anomalías y las envía al módulo de notificaciones push.

10. Onboarding

Facilita el registro inicial del usuario mediante la verificación de identidad, utilizando tecnologías como el reconocimiento facial o la autenticación biométrica. El módulo asegura que solo los usuarios verificados puedan registrarse y acceder al sistema, integrándose con el módulo de autenticación para validar las credenciales.

11. Auditoría (Audit)

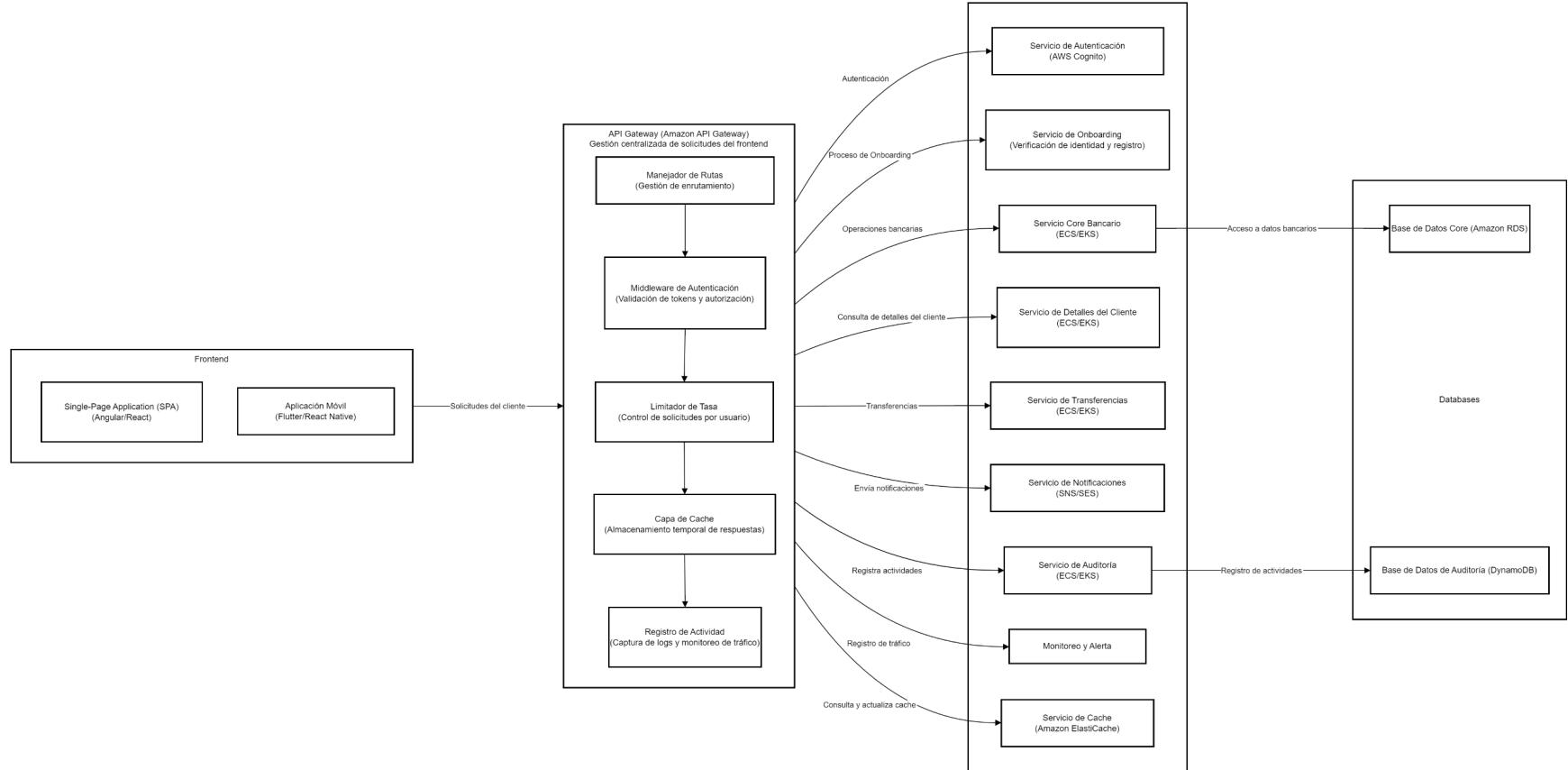
Captura y registra actividades críticas dentro de la aplicación, incluyendo el acceso, los cambios en la configuración y los errores significativos. Los registros de auditoría se almacenan para asegurar el cumplimiento normativo y para la revisión de incidentes en caso de sospecha de actividades no autorizadas.

Consideraciones de Seguridad y Buenas Prácticas

- Autenticación Segura: Implementar autenticación multifactor (MFA) para incrementar la seguridad en dispositivos móviles.
- Protección de Datos: Usar técnicas de cifrado para proteger los datos almacenados localmente y durante la transmisión.
- Monitoreo Proactivo: Configurar el módulo de monitoreo para supervisar métricas de rendimiento y registrar incidentes críticos en la aplicación.
- Cumplimiento Normativo: Garantizar que el módulo de auditoría cumpla con las regulaciones de seguridad de la información y privacidad.

BackEnd del Sistema de Banca por Internet

Módulo Componente API Gateway (Amazon API Gateway)



El **API Gateway** es el componente central encargado de gestionar todas las solicitudes provenientes de las aplicaciones frontend (SPA y aplicación móvil). Actúa como un intermediario que dirige el tráfico a los servicios backend adecuados, aplicando reglas de enrutamiento, autenticación, control de acceso y optimización de rendimiento.

Componentes del API Gateway

1. Manejador de Rutas (RouteHandler)

Gestiona el enrutamiento de solicitudes basado en la URL y el método HTTP (GET, POST, PUT, DELETE).

Se asegura de que cada solicitud sea dirigida al servicio backend correspondiente para ser procesada.

Verifica las rutas protegidas y las redirige si es necesario, en función de la política de acceso.

2. Middleware de Autenticación (AuthMiddleware)

Valida los tokens de autenticación que acompañan a cada solicitud, utilizando el servicio de autenticación (AWS Cognito).

Asegura que los usuarios tengan permisos adecuados para acceder a los recursos solicitados, aplicando políticas basadas en el rol del usuario.

Protege el sistema contra accesos no autorizados, verificando el estado de autenticación en cada solicitud.

3. Limitador de Tasa (RateLimiter)

Controla la cantidad de solicitudes permitidas por usuario para evitar abusos o sobrecargas en el sistema.

Aplica límites basados en la IP o en el usuario, con el fin de proteger la infraestructura contra ataques de denegación de servicio (DoS).

4. Capa de Cache (CachingLayer)

Implementa un almacenamiento temporal de las respuestas más frecuentes, mejorando el rendimiento y reduciendo la carga en los servicios backend.

Utiliza tecnologías como Amazon ElastiCache para administrar los datos en memoria, asegurando tiempos de respuesta rápidos para solicitudes recurrentes.

5. Registro de Actividad (Logging)

Captura logs detallados y métricas de tráfico en tiempo real, facilitando la supervisión y auditoría.

Se conecta al módulo de **Monitoreo y Alerta** para generar notificaciones en caso de incidentes críticos, como errores en la autenticación o intentos de sobrecarga.

Conexiones con Otros Servicios Backend

Servicio de Autenticación (AWS Cognito): Verifica la validez de los tokens y gestiona la autenticación de usuarios, garantizando que solo los usuarios autorizados accedan al sistema.

Servicio de Onboarding: Facilita el registro inicial y la verificación de identidad de los nuevos usuarios.

Servicio Core Bancario: Proporciona acceso a datos críticos, como saldos y movimientos de cuentas bancarias.

Servicio de Detalles del Cliente: Completa la información necesaria sobre el cliente, proporcionando detalles adicionales cuando se requieren.

Servicio de Transferencias: Gestiona las transferencias financieras, tanto internas como hacia otros bancos.

Servicio de Notificaciones: Envía alertas y notificaciones al usuario sobre actividades importantes, utilizando servicios como SNS o SES.

Servicio de Auditoría: Captura registros de actividades importantes para el cumplimiento de normativas.

Servicio de Cache: Almacena datos frecuentemente solicitados en memoria para mejorar la velocidad de acceso.

Monitoreo y Alerta: Supervisa el tráfico y el rendimiento del API Gateway, generando alertas en caso de detectar anomalías o caídas en el sistema.

Interacción con las Bases de Datos

Base de Datos Core (Amazon RDS): Utilizada para acceder a datos financieros y bancarios, como información de cuentas y transacciones.

Base de Datos de Auditoría (DynamoDB): Almacena los registros de auditoría generados por las solicitudes, proporcionando trazabilidad para el cumplimiento de normativas.

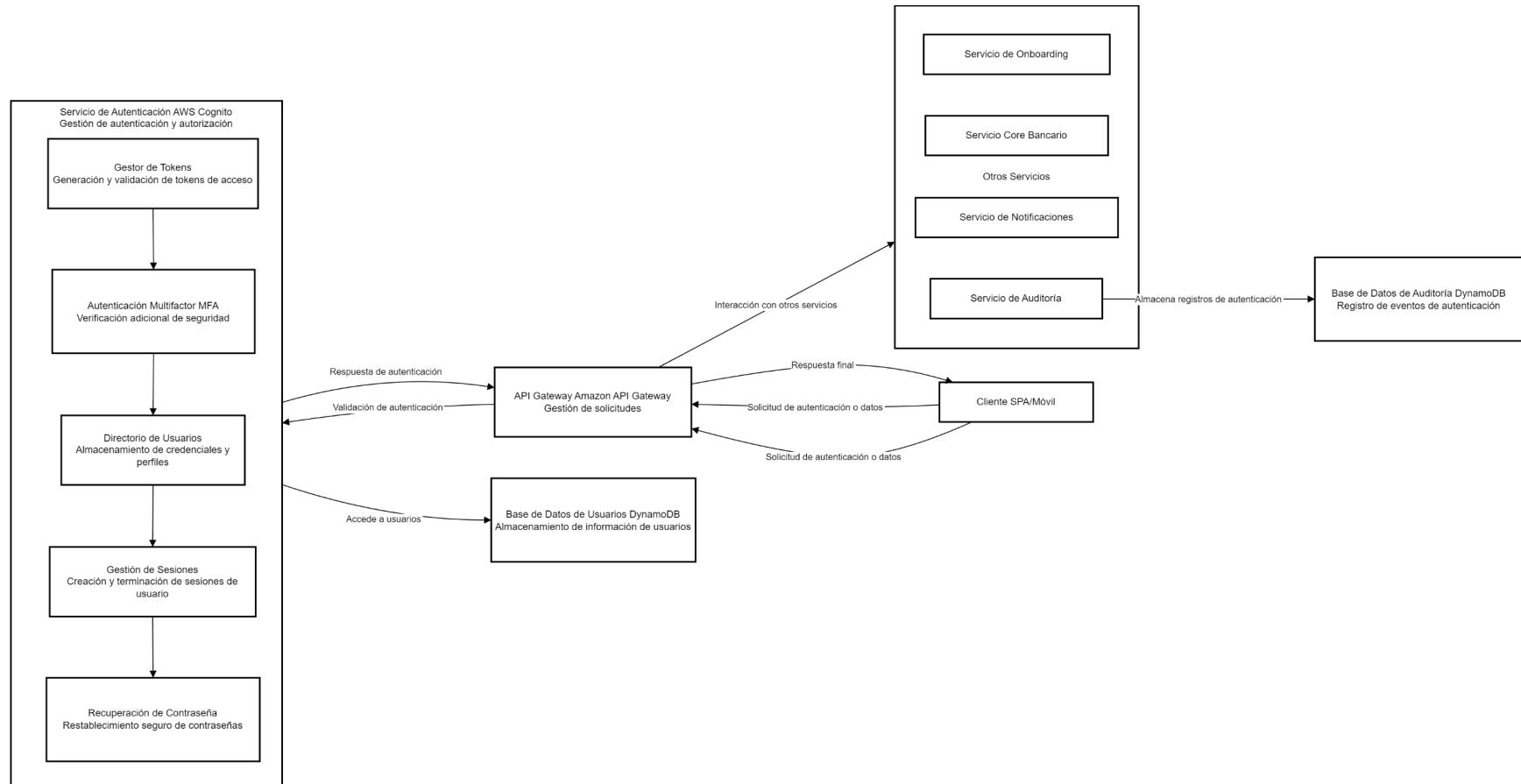
Características Técnicas

Alta Disponibilidad: El API Gateway distribuye el tráfico en varias zonas de disponibilidad para asegurar que el sistema siga funcionando incluso en caso de fallos.

Seguridad: Aplica autenticación y autorización integradas con AWS Cognito, protegiendo los recursos sensibles con políticas de acceso robustas.

Rendimiento Optimizado: Utiliza caching y limitación de tasa para mejorar la capacidad de respuesta y prevenir sobrecargas en los servicios backend.

Módulo Componente Servicio de Autenticación (AWS Cognito)



Descripción del Diagrama de Componentes: Servicio de Autenticación

El diagrama representa la arquitectura del **Servicio de Autenticación (AWS Cognito)** y su interacción con el **API Gateway (Amazon API Gateway)** y otros servicios agrupados. A continuación, se explican los elementos y flujos presentes en el diagrama:

1. Cliente (SPA/Móvil):

El cliente interactúa con el sistema a través de una aplicación web o móvil. Realiza solicitudes de autenticación, acceso a datos o diversas operaciones bancarias, enviando las solicitudes al **API Gateway**.

2. API Gateway (Amazon API Gateway):

Gestiona la enrutación de solicitudes hacia el **Servicio de Autenticación** o los servicios backend agrupados (Onboarding, Core Bancario, Notificaciones, Auditoría).

Realiza validaciones de seguridad y autenticación antes de permitir el acceso a los servicios backend.

3. Servicio de Autenticación (AWS Cognito):

Se encarga de la gestión de la autenticación y autorización de usuarios.

Gestor de Tokens: Genera y valida los tokens de acceso utilizados para la autenticación.

Autenticación Multifactor (MFA): Ofrece una capa adicional de seguridad mediante la verificación multifactor.

Directorio de Usuarios: Almacena y gestiona las credenciales y perfiles de los usuarios.

Gestión de Sesiones: Crea y termina sesiones de usuario según sea necesario.

Recuperación de Contraseña: Permite el restablecimiento seguro de contraseñas.

Realiza la validación de las solicitudes de autenticación y retorna la respuesta al API Gateway para autorizar o rechazar la solicitud.

4. Otros Servicios Agrupados:

Incluye servicios como **Onboarding**, **Core Bancario**, **Notificaciones** y **Auditoría**, que son invocados por el **API Gateway** según el tipo de operación solicitada.

Estos servicios manejan tareas específicas como la verificación de identidad durante el proceso de onboarding, la gestión de operaciones bancarias, el envío de notificaciones y el registro de actividades para auditoría.

5. Conexiones a las Bases de Datos:

Base de Datos de Usuarios (DynamoDB): Almacenamiento seguro de la información de los usuarios, utilizada por el Servicio de Autenticación para validar credenciales.

Base de Datos de Auditoría (DynamoDB): Registra los eventos de autenticación y actividades importantes del sistema para cumplir con los requisitos normativos.

6. Flujo de Solicitudes y Respuestas:

El flujo comienza cuando el cliente realiza una solicitud que es gestionada por el **API Gateway**.

El **API Gateway** valida la autenticación con el **Servicio de Autenticación** y, si es necesario, redirige la solicitud a otros servicios backend.

La respuesta se retorna al cliente después de ser procesada por el **API Gateway**, garantizando que todas las solicitudes sean seguras y controladas.

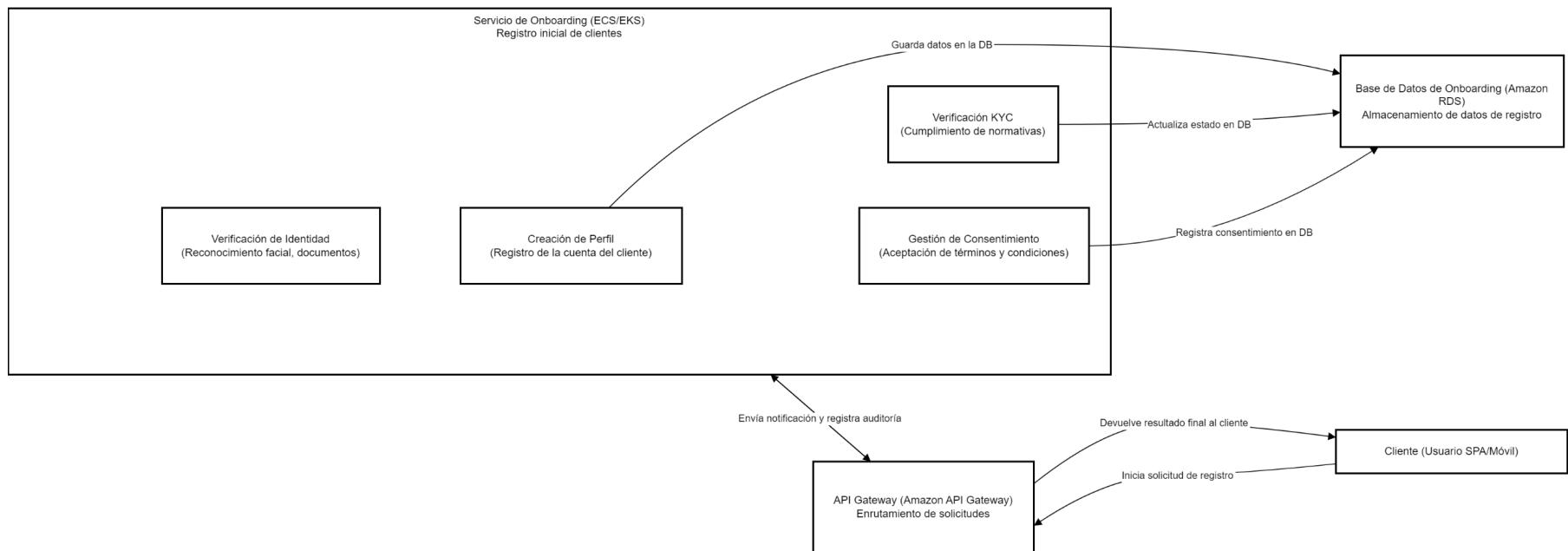
Consideraciones de Seguridad y Buenas Prácticas:

La autenticación se realiza utilizando el estándar **OAuth 2.0**, y se recomienda implementar **MFA** para mayor seguridad.

Los tokens generados por el **Servicio de Autenticación** deben ser almacenados de forma segura y tener un tiempo de expiración definido.

Es crucial registrar los eventos de autenticación en la **Base de Datos de Auditoría** para asegurar la trazabilidad y el cumplimiento normativo.

Módulo Componente Servicio OnBoarding



Servicio de Onboarding

El **Servicio de Onboarding** es una parte fundamental del sistema de banca por internet, encargado de gestionar el proceso de registro inicial de nuevos usuarios. A continuación, se detalla cada uno de los componentes y pasos involucrados en el proceso de onboarding:

1. Verificación de Identidad

Este componente se encarga de confirmar que el usuario es quien dice ser. Puede incluir métodos de verificación como el reconocimiento facial, la validación de documentos de identidad (pasaporte, licencia de conducir, etc.) y la comparación de datos biométricos.

La verificación de identidad es crucial para prevenir fraudes y asegurar que solo personas autorizadas accedan a los servicios del banco.

2. Creación de Perfil

Una vez que la identidad del usuario ha sido verificada, se procede a registrar la información personal en el sistema, creando un perfil que incluye detalles como el nombre, dirección, número de teléfono y otra información relevante.

El perfil del usuario es almacenado en la **Base de Datos de Onboarding** para futuras referencias y gestión.

3. Verificación KYC (Know Your Customer)

Este paso cumple con los requisitos regulatorios para conocer a los clientes del banco y prevenir actividades ilícitas, como el lavado de dinero. La verificación KYC puede incluir la comprobación de antecedentes, la verificación de fuentes de ingresos y la evaluación de riesgos del usuario.

Los resultados de la verificación se almacenan en la base de datos, asegurando un registro completo de las validaciones.

4. Gestión de Consentimiento

En este componente se gestionan los términos y condiciones que el usuario debe aceptar para poder utilizar los servicios del banco. Esto incluye las políticas de privacidad, los términos de uso y otros consentimientos necesarios para cumplir con las normativas.

Una vez que el usuario acepta los términos, el consentimiento es registrado en la base de datos.

5. Interacciones con Otros Servicios

Todas las solicitudes relacionadas con notificaciones, auditoría y autenticación pasan por el API Gateway, que se encarga de redirigirlas al servicio correspondiente.

Por ejemplo, las notificaciones se envían al Servicio de Notificaciones (SNS/SES), y los registros de auditoría son gestionados por el Servicio de Auditoría (ECS/EKS).

6. Flujo de Respuesta

Una vez completado el proceso de onboarding, el **Servicio de Onboarding** envía una respuesta final al **API Gateway**, que a su vez comunica el resultado al cliente.

Esto asegura que el usuario esté correctamente informado sobre el estado de su registro y pueda proceder a usar los servicios bancarios.

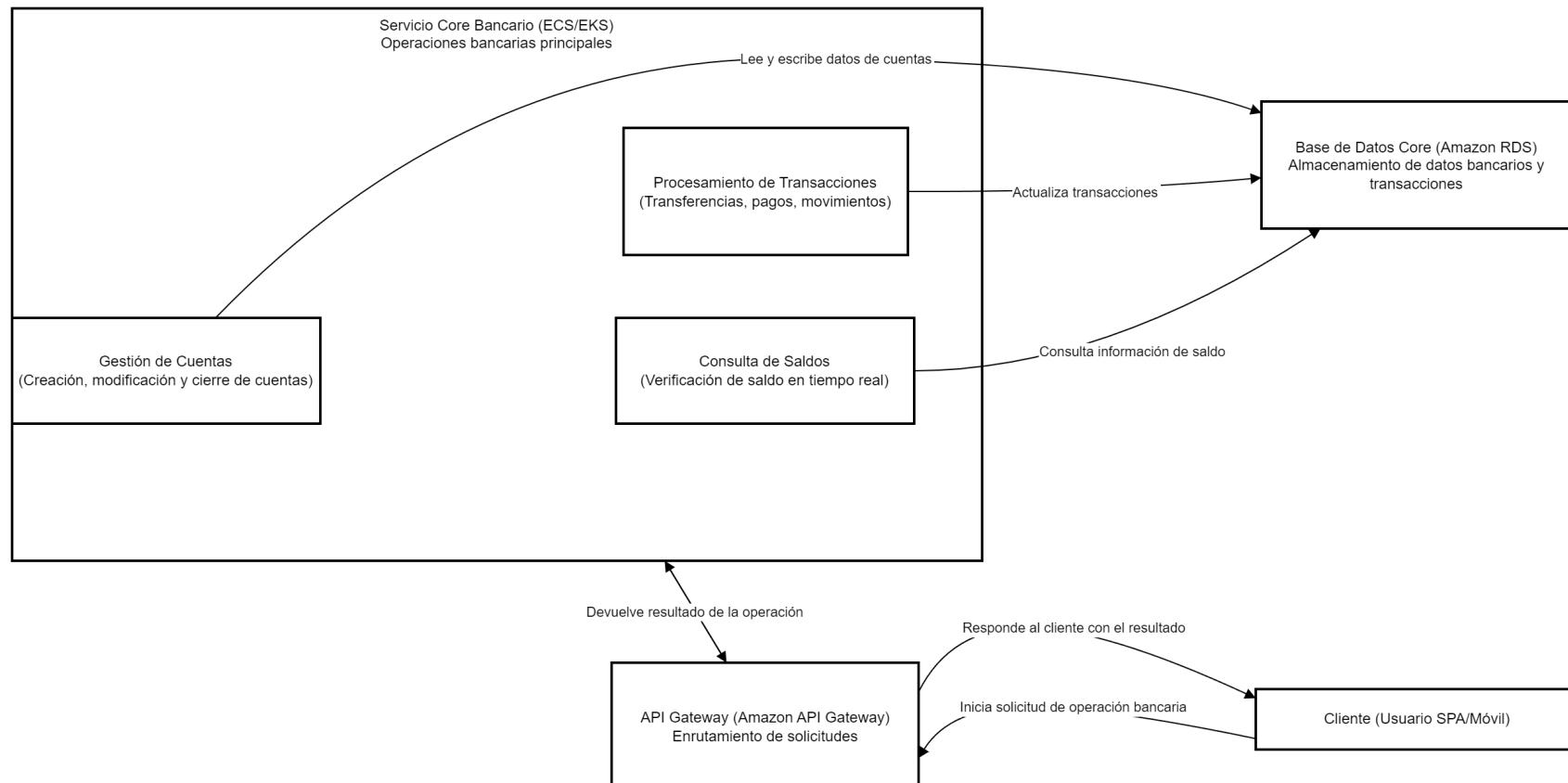
Consideraciones de Seguridad y Cumplimiento Normativo

El **Servicio de Onboarding** asegura que todas las etapas del proceso cumplan con las normativas de seguridad y regulaciones del sector financiero.

La verificación de identidad, la validación KYC y la gestión de consentimiento son componentes críticos para asegurar que el banco cumpla con las leyes contra el fraude y el lavado de dinero.

Módulo Componente Servicio Core Bancario

El **Servicio Core Bancario (ECS/EKS)** es el componente central del sistema de banca por internet encargado de gestionar las operaciones bancarias esenciales. Este servicio maneja todas las solicitudes relacionadas con cuentas, transacciones y consultas de saldo. A continuación, se detalla cada uno de los componentes:



1. Gestión de Cuentas

Este módulo es responsable de todas las operaciones relacionadas con la gestión de cuentas bancarias, como la creación de nuevas cuentas, modificación de datos, y el cierre de cuentas.

Toda la información sobre las cuentas se almacena en la **Base de Datos Core (Amazon RDS)**, garantizando la integridad y consistencia de los datos.

2. Procesamiento de Transacciones

Maneja el procesamiento de transferencias, pagos y otros movimientos financieros. Se asegura de que todas las transacciones se registren correctamente en la base de datos y de que se cumplan las normativas de trazabilidad.

Las transacciones se actualizan en la **Base de Datos Core**, donde se mantiene un registro detallado de cada movimiento realizado.

3. Consulta de Saldos

Permite a los usuarios verificar el saldo disponible en sus cuentas en tiempo real. El componente consulta la **Base de Datos Core** para obtener la información más actualizada sobre el saldo.

Esto es fundamental para las operaciones diarias, ya que asegura que los usuarios siempre tengan acceso a la información más reciente sobre sus fondos.

Flujos de Solicitud y Respuesta

Todas las solicitudes del Cliente pasan por el API Gateway, que redirige las solicitudes correspondientes al Servicio Core Bancario. El gateway asegura que las solicitudes sean correctamente enrutadas y gestionadas.

El Servicio Core Bancario procesa la solicitud y realiza las operaciones necesarias, interactuando con la Base de Datos Core para leer o actualizar la información.

Una vez completada la operación, el resultado se devuelve al API Gateway, que a su vez responde al Cliente.

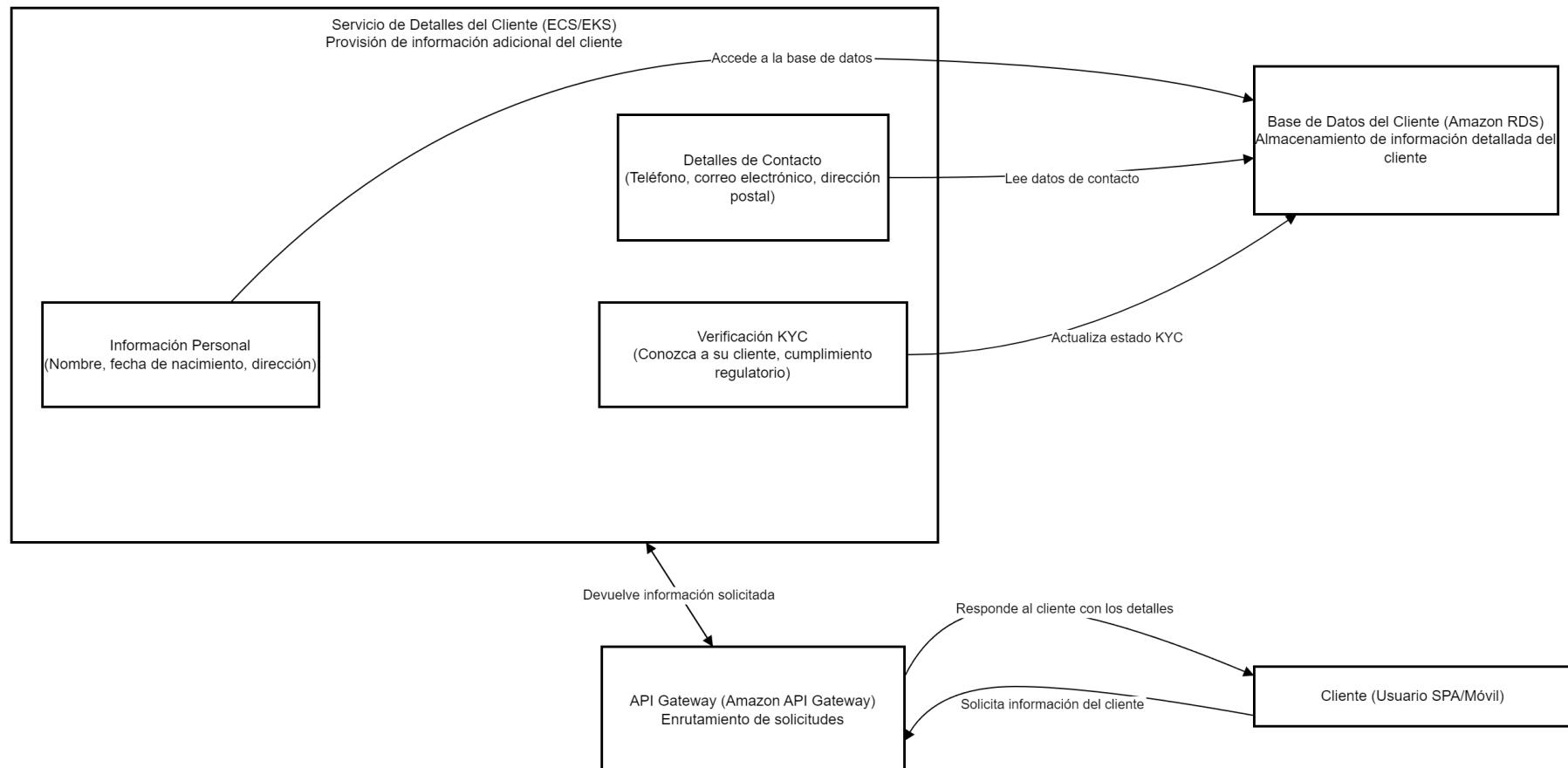
Consideraciones Técnicas y Buenas Prácticas

El Servicio Core Bancario utiliza servicios de contenedores en AWS (Amazon ECS o EKS) para asegurar una alta disponibilidad y escalabilidad.

Todas las operaciones están diseñadas para garantizar la integridad de los datos, con medidas de control para evitar inconsistencias en las transacciones.

Se implementan mecanismos de auditoría para registrar todas las operaciones realizadas, lo que es crucial para cumplir con las normativas del sector financiero.

Módulo Componente Servicio de Detalles del Cliente



El **Servicio de Detalles del Cliente (ECS/EKS)** complementa la información del cliente con datos adicionales que pueden no estar disponibles en la base de datos principal. Es esencial para proporcionar detalles completos del cliente en situaciones que requieren información personal o cumplimiento normativo (KYC). A continuación, se describen sus componentes:

1. Información Personal

Proporciona datos básicos del cliente, como nombre, fecha de nacimiento y dirección. Esta información es consultada y mantenida en la **Base de Datos del Cliente (Amazon RDS)**.

La actualización de esta información se lleva a cabo bajo políticas estrictas de integridad de datos y cumplimiento normativo.

2. Detalles de Contacto

Incluye datos de contacto del cliente, tales como número de teléfono, correo electrónico y dirección postal. Estos detalles son fundamentales para la comunicación y el envío de notificaciones.

El servicio verifica y mantiene actualizada esta información para asegurar que las notificaciones lleguen al cliente de manera efectiva.

3. Verificación KYC (Know Your Customer)

Realiza comprobaciones para garantizar que el cliente cumpla con las normativas regulatorias del sector financiero, como la prevención de lavado de dinero.

Los resultados de las verificaciones KYC se almacenan en la **Base de Datos del Cliente**, asegurando que estén siempre actualizados.

Flujos de Solicitud y Respuesta

Todas las solicitudes de información del cliente pasan por el **API Gateway**, que se encarga de redirigir la solicitud al **Servicio de Detalles del Cliente**. Esto asegura que solo los usuarios autorizados puedan acceder a la información del cliente.

El servicio consulta la **Base de Datos del Cliente** para obtener los datos necesarios y realiza las verificaciones KYC antes de devolver la información al **API Gateway**.

Finalmente, el **API Gateway** responde al cliente con la información solicitada, manteniendo el flujo seguro y controlado.

Consideraciones Técnicas y Buenas Prácticas

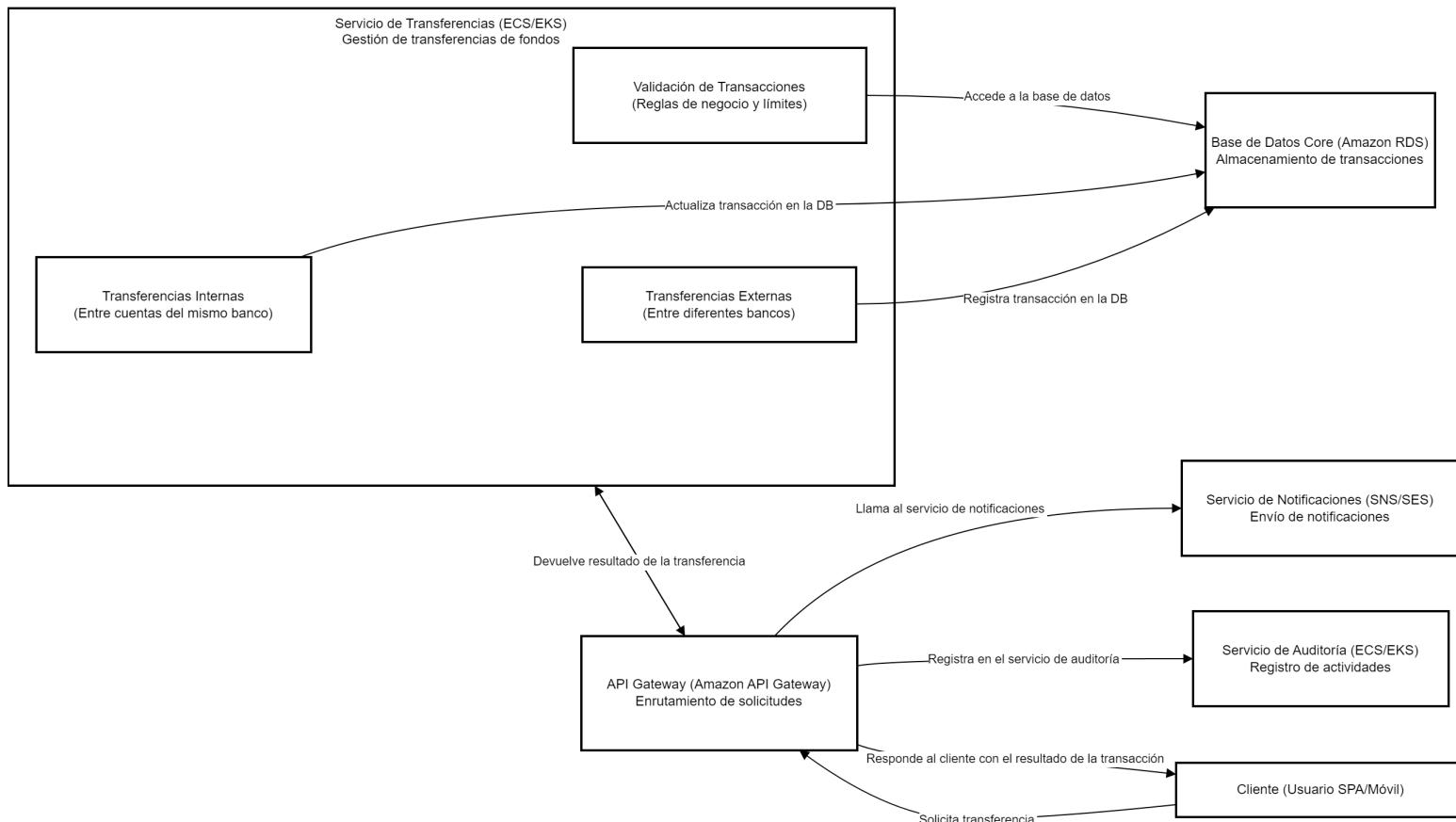
El servicio está diseñado para ser escalable y garantizar la alta disponibilidad utilizando contenedores en AWS (Amazon ECS o EKS).

La seguridad de los datos es prioritaria, implementando controles de acceso y monitoreo para prevenir el acceso no autorizado.

El cumplimiento normativo es gestionado de forma estricta para asegurar que todas las verificaciones KYC se realicen de manera efectiva y oportuna.

Módulo Componente Servicio de Transferencias (ECS/EKS)

El **Servicio de Transferencias (ECS/EKS)** es responsable de manejar las transferencias de fondos dentro del sistema bancario, tanto internas como externas. El flujo del proceso y las interacciones a través del **API Gateway** aseguran que todas las comunicaciones sean seguras y controladas.



El Servicio de Transferencias (ECS/EKS) se encarga de la gestión completa de las transferencias de fondos, ya sean internas (dentro del mismo banco) o externas (hacia otros bancos), siguiendo las normativas del sector financiero.

Transferencias Internas

Gestiona las transacciones entre cuentas del mismo banco. Estos movimientos se realizan de forma interna y actualizan inmediatamente los registros en la Base de Datos Core (Amazon RDS) para reflejar los saldos actualizados.

Transferencias Externas

Maneja las transacciones hacia otros bancos, lo cual involucra la integración con redes de pagos interbancarios para asegurar que la transacción se procese correctamente.

Los registros de estas transacciones se mantienen en la Base de Datos Core, garantizando la trazabilidad y conformidad.

Validación de Transacciones

Antes de procesar cualquier transferencia, se verifica que la transacción cumpla con las reglas de negocio, como límites de monto, disponibilidad de fondos y políticas de seguridad.

Esta validación incluye la consulta al saldo actual en la Base de Datos Core.

Flujos de Solicitud y Respuesta

El proceso comienza cuando el Cliente inicia una solicitud de transferencia a través del API Gateway, el cual se encarga de redirigirla al Servicio de Transferencias.

El servicio realiza las validaciones necesarias y procesa la transferencia. Luego, actualiza los registros en la base de datos.

Las notificaciones sobre la transacción y el registro de auditoría se gestionan también a través del API Gateway, lo cual centraliza todas las comunicaciones externas.

Buenas Prácticas y Seguridad

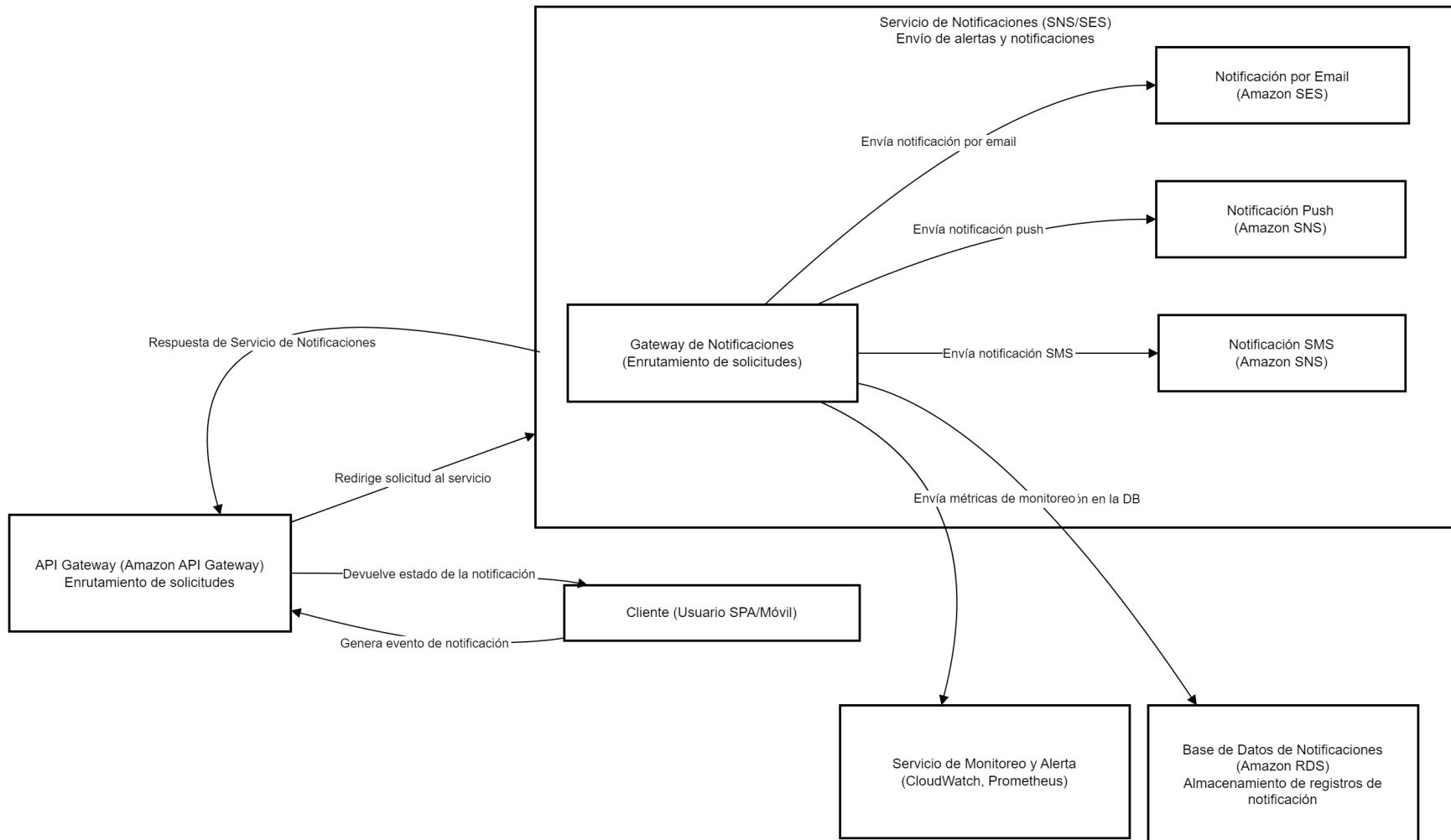
El uso del API Gateway para canalizar todas las interacciones asegura que no haya comunicaciones directas no controladas entre servicios.

El servicio de transferencias implementa políticas de auditoría rigurosas para todas las operaciones, lo que garantiza el cumplimiento normativo y la seguridad en las transacciones.

Las validaciones de transacciones se realizan antes de autorizar cualquier transferencia para prevenir fraudes y errores.

Módulo Componente Servicio de Notificaciones

El Servicio de Notificaciones (SNS/SES) gestiona la entrega de alertas y notificaciones a los usuarios mediante diferentes canales. La adición de un Gateway de Notificaciones interno optimiza el proceso al decidir dinámicamente el canal de entrega más adecuado.



1. Gateway de Notificaciones

Función principal: Actúa como un punto central dentro del servicio para enrutar las solicitudes de notificación al canal adecuado, según el tipo de evento y las preferencias del usuario.

Decisión del canal: Evalúa el tipo de notificación solicitada y la configuración de preferencia del usuario para determinar si debe enviarse a través de email, notificación push o SMS.

Auditoría y Registro: Antes de enviar la notificación, registra la solicitud en la **Base de Datos de Notificaciones**, lo cual permite un seguimiento completo.

2. Canales de Notificación

Email Notification (Amazon SES): Elige este canal para notificaciones detalladas y formales que requieren documentación o seguimiento.

Push Notification (Amazon SNS): Utilizado para alertas en tiempo real en dispositivos móviles, proporcionando actualizaciones rápidas y contextuales.

SMS Notification (Amazon SNS): Ideal para notificaciones urgentes cuando el acceso a internet podría ser limitado.

Flujos de Solicitud y Respuesta

1. Inicio del Flujo:

El **Cliente** genera un evento que requiere una notificación, y la solicitud es redirigida por el **API Gateway** al **Servicio de Notificaciones**.

2. Enrutamiento:

El **Gateway de Notificaciones** determina el canal de notificación y procede a enviar la alerta a través de los canales configurados.

3. Registro y Monitoreo:

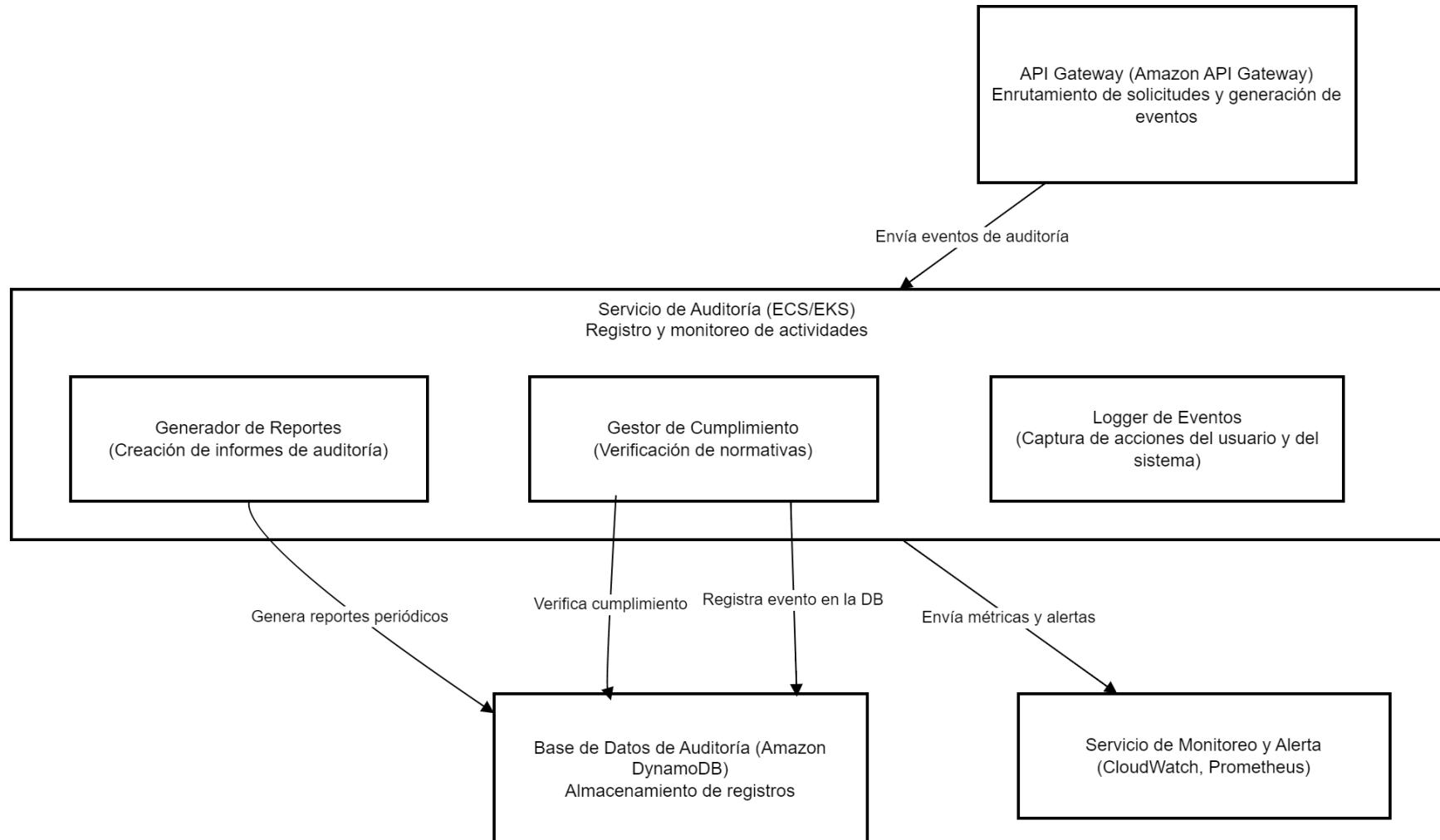
Cada evento de notificación se registra en la **Base de Datos de Notificaciones** para garantizar la trazabilidad.

El servicio también reporta métricas al **Servicio de Monitoreo y Alerta**, permitiendo una supervisión constante y la detección de posibles errores.

4. Respuesta al Cliente:

- Despues de procesar la solicitud, el **API Gateway** devuelve el estado de la notificación al **Cliente**, confirmando si se envió correctamente.
- **Consideraciones Técnicas y Buenas Prácticas**
- **Modularidad:** La implementación del **Gateway de Notificaciones** asegura que las decisiones de enrutamiento sean manejadas de manera centralizada, lo que facilita la adición de nuevos canales en el futuro.
- **Monitoreo Proactivo:** La integración con el servicio de monitoreo permite detectar fallos en el envío de notificaciones y generar alertas automáticamente si se supera un umbral de errores.
- **Auditoría Completa:** El registro de todas las solicitudes en la base de datos facilita el cumplimiento normativo y la detección de problemas en la entrega de mensajes.

Módulo Componente Servicio de Auditoría (ECS/EKS)



El **Servicio de Auditoría (ECS/EKS)** es fundamental para asegurar la trazabilidad y el cumplimiento normativo en el sistema de banca por internet. Se encarga de registrar todas las actividades importantes y mantener un historial detallado de los eventos que ocurren en el sistema.

Componentes Internos

Logger de Eventos:

Captura todas las actividades relevantes del sistema, incluyendo acciones realizadas por los usuarios y procesos automáticos. Los eventos registrados se almacenan en la **Base de Datos de Auditoría (Amazon DynamoDB)** para mantener un historial seguro y auditável.

Gestor de Cumplimiento:

Verifica que los registros cumplan con las normativas internas y externas aplicables, generando alertas cuando se detectan actividades sospechosas o fuera de lo común.

Es un componente clave para asegurar que las políticas de la empresa se cumplan en todo momento.

Generador de Reportes:

Crea informes periódicos de auditoría para revisiones internas y auditorías externas.

Los reportes incluyen métricas de actividad, alertas generadas, y el estado general del cumplimiento normativo.

Flujos de Registro y Auditoría

Captura de Eventos:

El **API Gateway** redirige eventos relevantes al **Servicio de Auditoría**, donde el **Logger de Eventos** los captura y almacena en la **Base de Datos de Auditoría**.

Verificación y Cumplimiento:

El **Gestor de Cumplimiento** analiza los registros de la base de datos para verificar que todas las acciones se adhieren a las normativas establecidas.

Las alertas se generan si se detectan actividades inusuales o que violan las políticas del sistema.

Generación de Reportes:

El **Generador de Reportes** elabora informes detallados que se almacenan en la base de datos para ser revisados por los auditores.

Monitoreo Integrado:

El **Servicio de Auditoría** también envía métricas al **Servicio de Monitoreo y Alerta** para mantener una supervisión en tiempo real del sistema y detectar rápidamente cualquier incidencia.

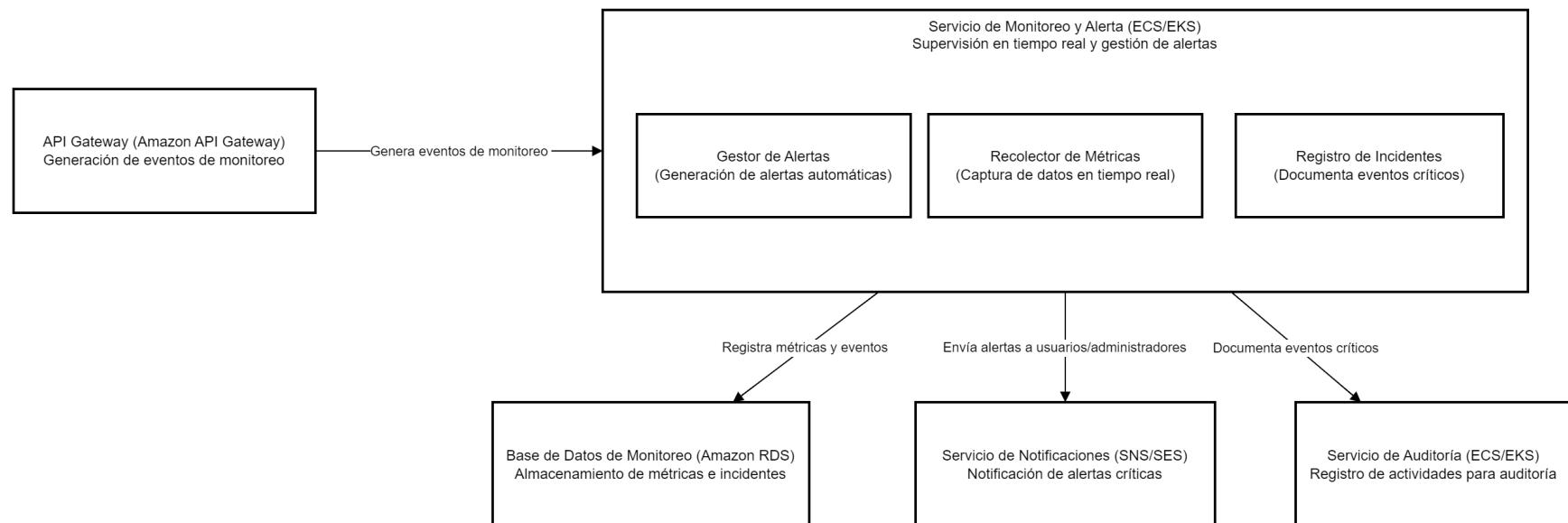
Consideraciones Técnicas y Buenas Prácticas

Seguridad de los Datos: Los registros de auditoría deben ser inmutables para garantizar que la información no sea alterada o eliminada.

Automatización del Cumplimiento: La verificación automática de normativas ayuda a identificar problemas antes de que se conviertan en incidentes graves.

Reportes Periódicos: Generar informes regulares permite a la administración mantenerse al tanto del estado del cumplimiento normativo y tomar medidas proactivas si es necesario.

Módulo Componente Servicio de Auditoría (ECS/EKS)



El **Servicio de Monitoreo y Alerta (ECS/EKS)** es una pieza clave para garantizar el estado óptimo del sistema de banca por internet. Este componente se encarga de la supervisión continua del rendimiento, la captura de incidentes y la generación de alertas automáticas para notificar problemas.

Componentes Internos

Recolector de Métricas:

Captura en tiempo real las métricas de rendimiento y errores, tales como tiempos de respuesta, uso de CPU, memoria, y latencia.

Permite detectar rápidamente cualquier desviación respecto al comportamiento normal del sistema.

Gestor de Alertas:

Genera alertas automáticas cuando las métricas alcanzan umbrales críticos. Las alertas pueden ser enviadas por email, SMS o push notifications.

Prioriza las alertas según su severidad y envía notificaciones al **Servicio de Notificaciones (SNS/SES)** para informar a los administradores.

Registro de Incidentes:

Documenta todos los eventos críticos y problemas detectados en la **Base de Datos de Monitoreo**.

Las actividades relevantes también se envían al **Servicio de Auditoría** para mantener un historial de incidentes que cumple con las normativas.

Flujos de Monitoreo y Alerta

Generación de Eventos:

El **API Gateway** captura y envía eventos de monitoreo generados por las acciones de los usuarios o del propio sistema. Estos eventos se procesan en el **Servicio de Monitoreo y Alerta**.

Captura y Almacenamiento de Métricas:

El **Recolector de Métricas** procesa los datos en tiempo real y los almacena en la **Base de Datos de Monitoreo (Amazon RDS)** para análisis y seguimiento posterior.

Notificación de Alertas:

Cuando se detectan problemas críticos, el **Gestor de Alertas** envía notificaciones a los administradores a través del **Servicio de Notificaciones** para que tomen las acciones correspondientes.

Registro de Incidentes:

Todos los incidentes capturados por el **Gestor de Alertas** se documentan en el **Registro de Incidentes**, lo que también permite enviar estos eventos al **Servicio de Auditoría** para cumplir con los requisitos normativos.

Consideraciones Técnicas y Buenas Prácticas

Monitoreo en Tiempo Real: Asegurar la captura de métricas relevantes y configurar umbrales adecuados para evitar falsas alarmas.

Automatización de Respuestas: Configurar respuestas automáticas para ciertos eventos críticos (auto-healing) para minimizar el tiempo de inactividad.

Integración con Auditoría: El registro de incidentes debe ser enviado al **Servicio de Auditoría** para mantener la trazabilidad y facilitar la investigación en caso de auditorías externas.

Módulo Componente Servicio de Cache (Amazon ElastiCache)

El **Servicio de Cache (Amazon ElastiCache)** es una solución en la nube que proporciona almacenamiento en memoria para acelerar las respuestas del sistema de banca por internet. Utiliza tecnologías como Redis o Memcached para almacenar datos temporalmente y mejorar el rendimiento general, reduciendo la carga en la base de datos principal.

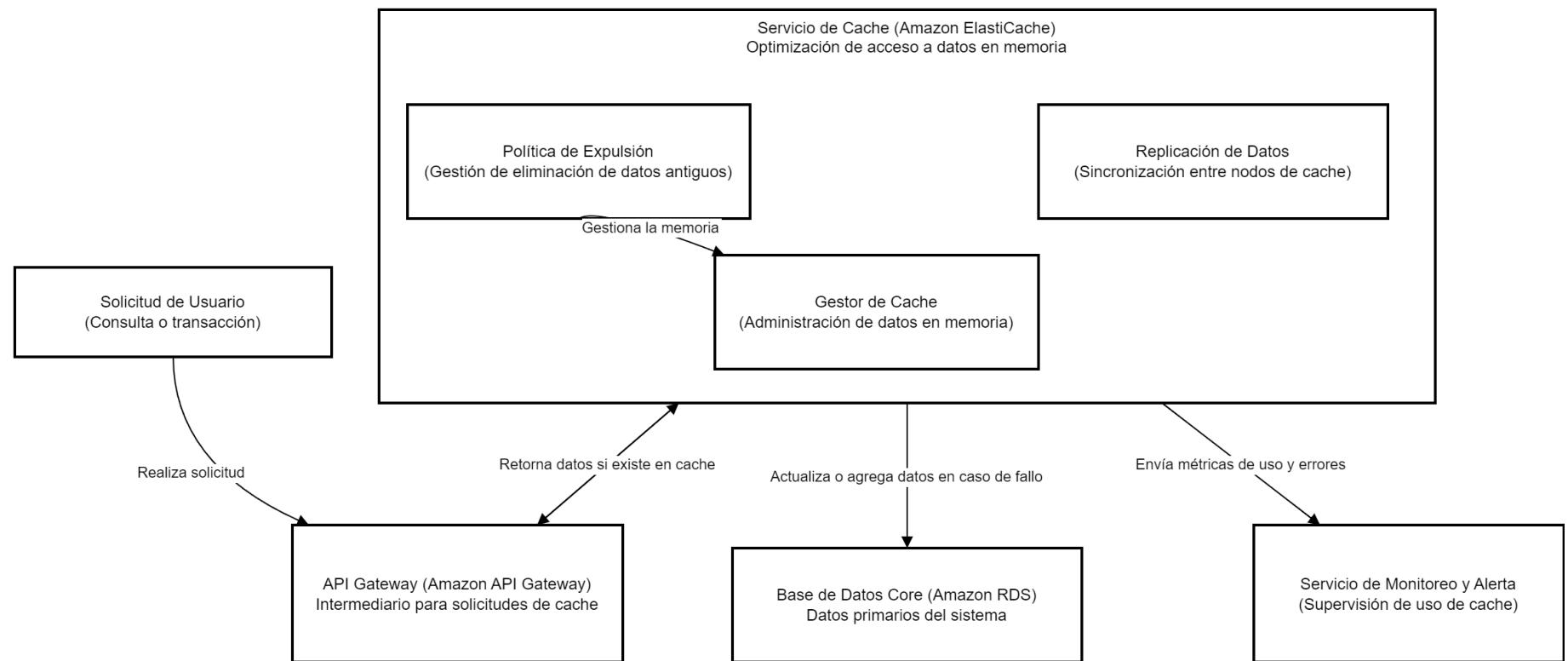
1. ¿Por Qué Usar Cache?

La cache se utiliza para almacenar datos que son solicitados con frecuencia y que, en algunos casos, pueden cambiar, como el saldo de una cuenta. Si bien el saldo de una cuenta puede variar a lo largo del día, el uso de cache puede ser beneficioso en los siguientes escenarios:

Consultas Frecuentes en Corto Plazo: Si un usuario revisa su saldo varias veces en pocos minutos, la cache puede almacenar el valor temporalmente para evitar múltiples consultas a la base de datos, optimizando el rendimiento.

Datos Temporales Durante Procesos: En algunos procesos, como la validación de transacciones, la cache puede almacenar saldos temporales para ofrecer una experiencia de usuario más rápida.

Actualización Automática: La cache puede configurarse para invalidar o actualizar el dato automáticamente después de una transacción que afecte el saldo, garantizando la consistencia.



Componentes Internos del Servicio de Cache

Gestor de Cache (CacheManager):

Administra los datos en memoria, determinando si el dato solicitado ya está en la cache. Si el dato está en la cache, se devuelve inmediatamente al **API Gateway**. Si no, se obtiene de la **Base de Datos Core** y se almacena en la cache.

Replicación de Datos (DataReplication):

Sincroniza los datos en múltiples nodos de cache para asegurar la disponibilidad y redundancia.

Permite que el sistema continúe funcionando sin problemas en caso de falla de un nodo.

Política de Expulsión (EvictionPolicy):

Administra la eliminación de datos antiguos para liberar espacio en memoria.

Utiliza políticas como **Least Recently Used (LRU)** o **Time-To-Live (TTL)** para eliminar datos no utilizados o expirados.

Flujos de Trabajo del Servicio de Cache

Solicitud del Usuario:

Cuando un usuario realiza una consulta (como el saldo de una cuenta), el **API Gateway** verifica primero si el dato está en la cache. Si está, el dato se devuelve de inmediato, mejorando el tiempo de respuesta.

Actualización en Caso de Fallo en la Cache:

Si el dato no está en la cache (fallo de cache), el **CacheManager** lo obtiene de la **Base de Datos Core**, lo almacena en la cache, y luego responde la solicitud.

Sincronización y Expulsión de Datos:

La **Replicación de Datos** asegura que todos los nodos de cache estén sincronizados. La **Política de Expulsión** administra la memoria para evitar el almacenamiento de datos innecesarios.

Monitoreo y Métricas:

Las métricas sobre el uso de la cache y los errores se envían al **Servicio de Monitoreo y Alerta** para asegurar un funcionamiento óptimo.

Consideraciones Técnicas

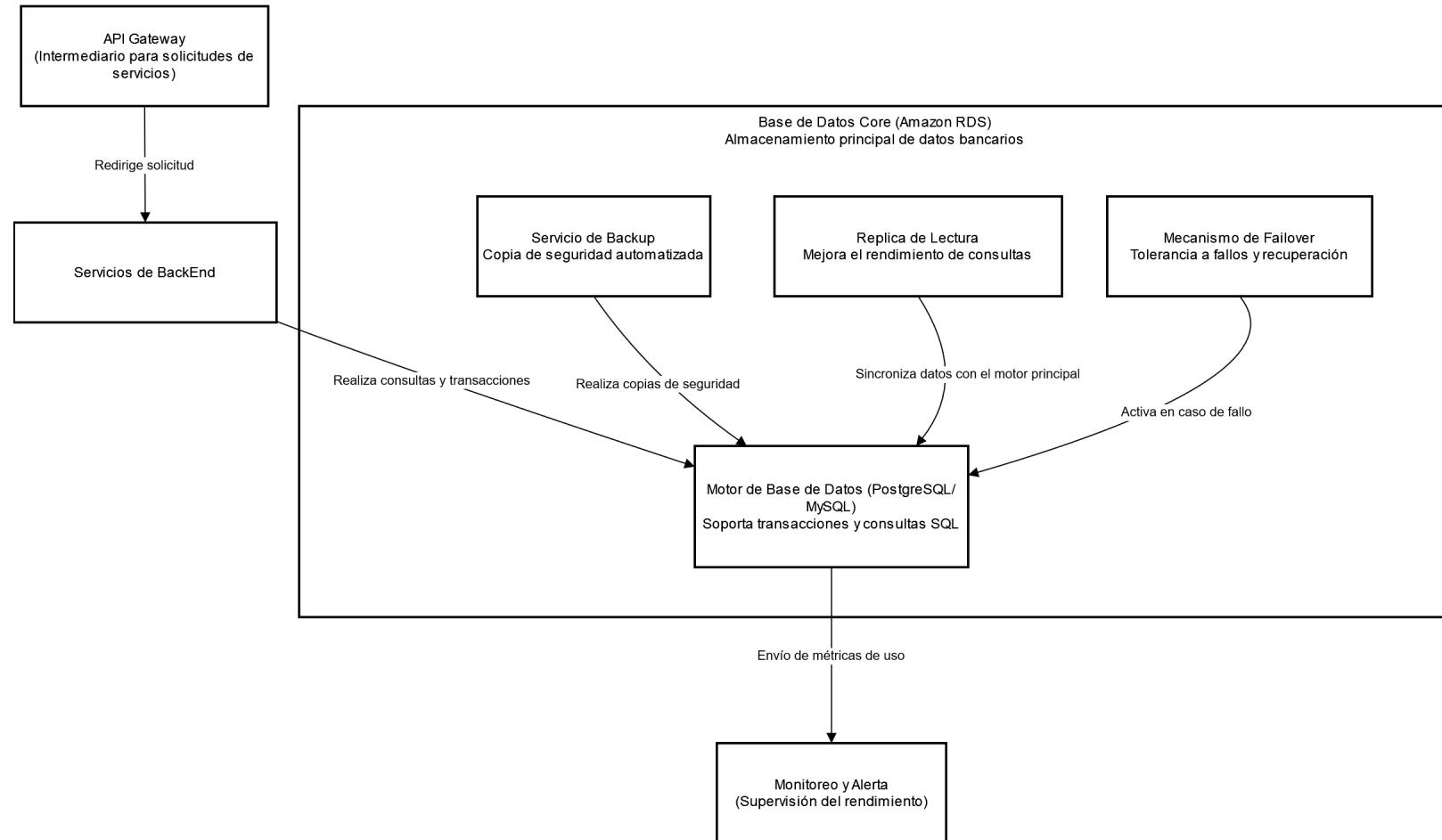
Configurar el Tiempo de Expiración Adecuado: Es importante definir tiempos cortos de expiración para datos como el saldo de cuentas, a fin de minimizar el riesgo de mostrar datos desactualizados.

Evitar Datos Obsoletos: La política de cache debe incluir reglas de invalidación para mantener los datos precisos y frescos.

Monitoreo Activo: Utilizar herramientas de monitoreo como **CloudWatch** para detectar problemas y ajustar la configuración del servicio.

Base de datos del Sistema de Banca por Internet

Base de Datos Core (Amazon RDS)



La **Base de Datos Core (Amazon RDS)** es el componente central para el almacenamiento de datos críticos en el sistema bancario. Se encarga de gestionar la información bancaria principal, como cuentas, transacciones y otros registros financieros esenciales. La arquitectura de la base de datos está diseñada para garantizar alta disponibilidad, rendimiento óptimo y recuperación ante desastres. A continuación, se detallan sus componentes y funcionalidades clave:

Motor de Base de Datos (PostgreSQL/MySQL)

Utiliza un motor de base de datos relacional (como PostgreSQL o MySQL), lo que permite realizar transacciones y consultas SQL complejas.

Es el núcleo del almacenamiento, donde se llevan a cabo las operaciones de lectura y escritura de datos.

Servicio de Backup

Realiza copias de seguridad automáticas de los datos en intervalos regulares, garantizando que la información esté protegida y disponible en caso de pérdida o corrupción de datos.

Las copias de seguridad se gestionan automáticamente, y se pueden restaurar en cualquier momento para recuperar el estado anterior de la base de datos.

Replica de Lectura

La réplica de lectura se sincroniza continuamente con el motor de base de datos principal, lo que permite manejar las consultas de lectura con un alto rendimiento y reducir la carga en el servidor principal.

Es especialmente útil para operaciones intensivas de lectura, como la generación de reportes y consultas frecuentes.

Mecanismo de Failover

Proporciona tolerancia a fallos mediante la activación de un nodo de reserva en caso de que el motor principal experimente un problema.

El failover automático asegura que el servicio se mantenga disponible sin interrupciones significativas, redirigiendo las solicitudes al nodo de respaldo.

Interacción con otros Servicios

El **API Gateway** redirige las solicitudes que requieren acceso a la base de datos al **Servicio Core Bancario**, el cual se encarga de gestionar las operaciones de consulta y actualización de datos.

El **Servicio de Monitoreo y Alerta** recibe métricas de uso y rendimiento del motor de base de datos para asegurar que el sistema funcione de manera óptima.

Características Adicionales

Alta Disponibilidad: Implementación en múltiples zonas de disponibilidad para evitar interrupciones del servicio.

Escalabilidad Automática: Capacidad para ajustar los recursos de la base de datos en función de la carga y el tráfico.

Encriptación de Datos: Los datos almacenados están encriptados para cumplir con los estándares de seguridad y privacidad.

Seguridad y Cifrado de Datos:

La Base de Datos Core utiliza cifrado en reposo mediante AWS KMS (Key Management Service) para proteger los datos sensibles.

Se implementa cifrado en tránsito utilizando TLS (Transport Layer Security) para asegurar que los datos transferidos entre los servicios y la base de datos estén protegidos.

Se configurarán políticas de acceso basadas en roles (IAM roles) para asegurar que solo los servicios autorizados puedan acceder a la base de datos.

Esquemas de Alta Disponibilidad y Failover:

Se configura una réplica de lectura en una región diferente para mejorar el rendimiento de las consultas y proporcionar tolerancia a fallos.

En caso de un fallo en la instancia principal, el sistema puede realizar un failover automático a la réplica de lectura, asegurando la continuidad del servicio.

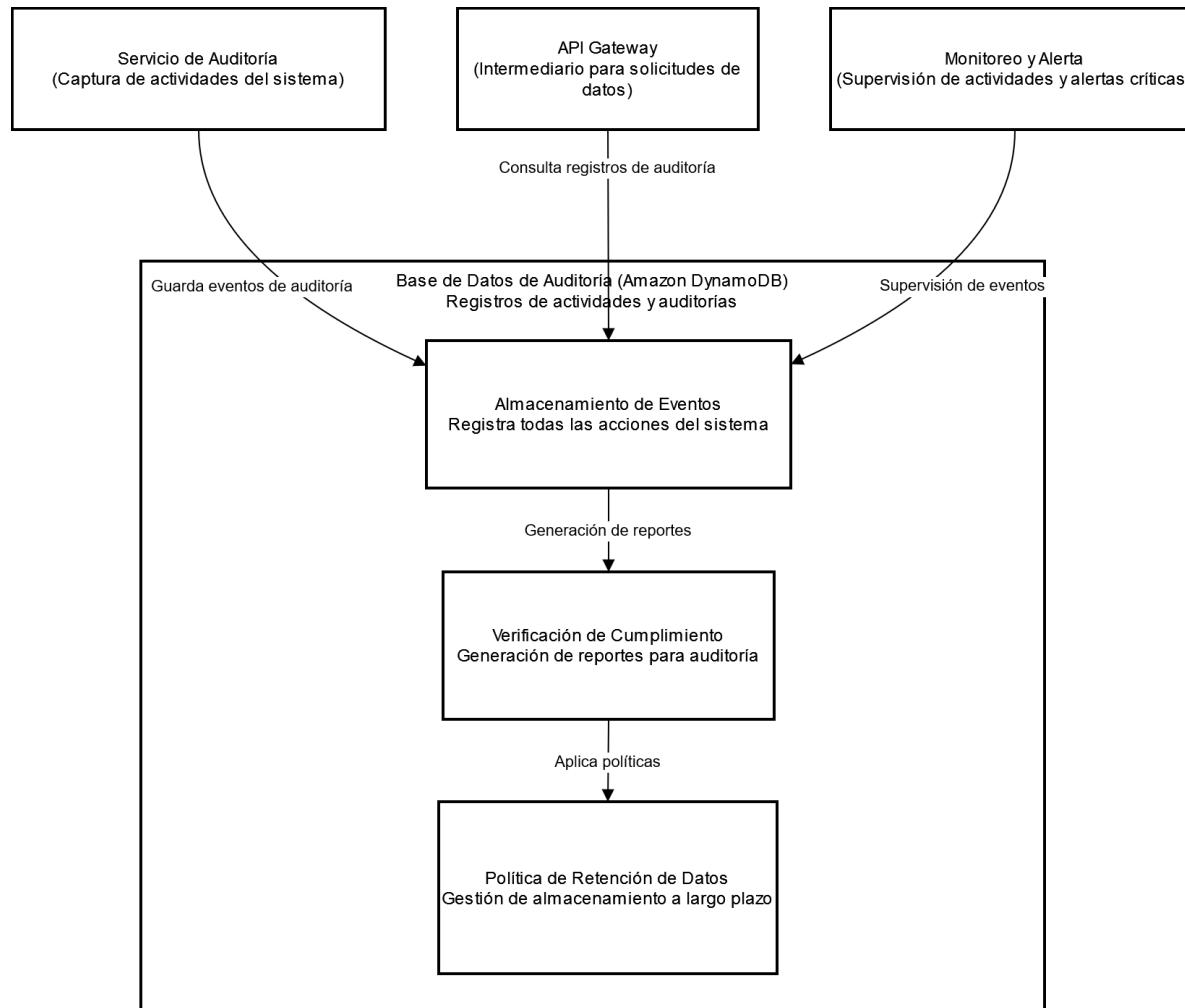
Procesos de Backup y Recuperación ante Desastres:

Amazon RDS realiza copias de seguridad automáticas diarias de la base de datos, incluyendo la replicación de logs de transacciones para restaurar el estado exacto.

Se programan backups manuales adicionales antes de realizar cualquier cambio significativo en la base de datos.

En caso de desastre, se utiliza la funcionalidad de recuperación puntual (point-in-time recovery) para restaurar la base de datos a un estado previo.

Base de Datos Auditoría (Amazon DynamoDB)



La **Base de Datos de Auditoría (Amazon DynamoDB)** es utilizada para registrar y almacenar todas las acciones y eventos relevantes que ocurren en el sistema. Su diseño garantiza que se cumplan las normativas de cumplimiento y que se pueda realizar un seguimiento preciso de las actividades para fines de auditoría.

1. Almacenamiento de Eventos

Este componente es responsable de registrar todas las acciones que se realizan en el sistema, incluyendo operaciones de usuarios y eventos críticos. Cada evento se almacena con detalles relevantes como el tipo de acción, el usuario involucrado, y la hora en que ocurrió.

El uso de **DynamoDB** permite almacenar grandes volúmenes de datos y realizar búsquedas rápidas para fines de auditoría.

2. Verificación de Cumplimiento

Este módulo se encarga de generar reportes periódicos de las actividades registradas, asegurando que el sistema cumpla con los requisitos de normativas y políticas internas.

Los reportes son utilizados por el equipo de auditoría para validar el cumplimiento de las normas y detectar cualquier actividad inusual.

3. Política de Retención de Datos

Los datos almacenados en la base de datos de auditoría se gestionan según las políticas de retención definidas, eliminando o archivando registros antiguos para optimizar el almacenamiento y cumplir con las regulaciones de privacidad.

4. Interacción con Otros Servicios

El **Servicio de Auditoría** registra las actividades y envía los eventos al **Almacenamiento de Eventos**.

El **API Gateway** permite acceder a los registros de auditoría cuando se necesiten para generar reportes o realizar verificaciones de seguridad.

El **Servicio de Monitoreo y Alerta** supervisa los eventos registrados y puede generar alertas en caso de actividades inusuales o potencialmente maliciosas.

5. Seguridad y Cifrado de Datos:

Los datos en DynamoDB están cifrados en reposo utilizando AWS KMS para proteger la integridad de los registros de auditoría.

Las comunicaciones entre los servicios y DynamoDB están cifradas en tránsito mediante TLS.

Se implementan políticas de acceso granular en IAM para garantizar que solo los servicios autorizados puedan leer o escribir en la base de datos.

6. Esquemas de Alta Disponibilidad y Failover:

DynamoDB ofrece alta disponibilidad por defecto, replicando los datos en múltiples zonas de disponibilidad en la misma región.

En caso de un fallo en una zona de disponibilidad, el servicio continúa operando utilizando las réplicas en las otras zonas.

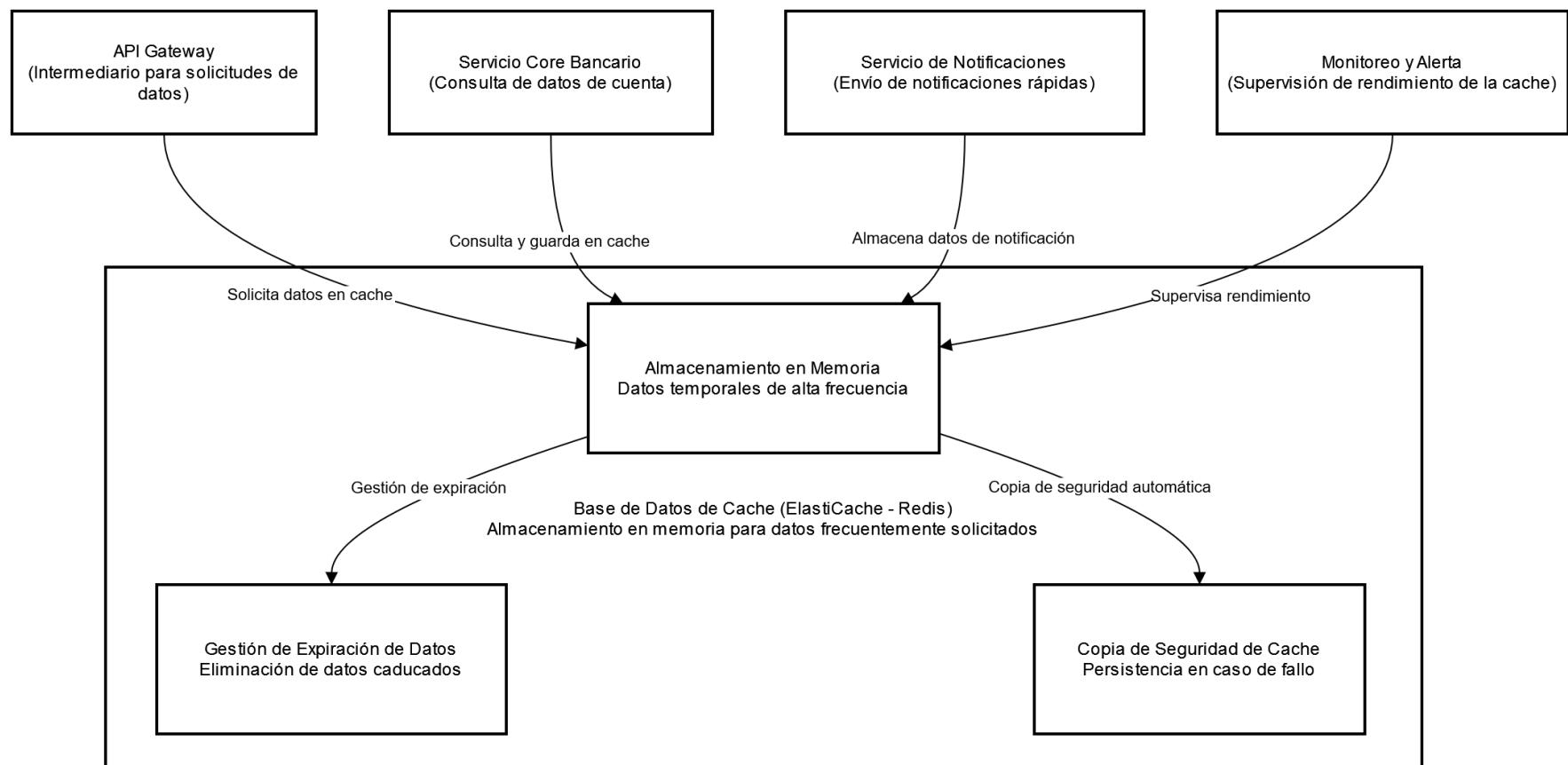
7. Procesos de Backup y Recuperación ante Desastres:

Se configuran backups automáticos con Amazon DynamoDB On-Demand, que permiten realizar copias de seguridad completas en cualquier momento.

Los backups se almacenan en Amazon S3 para garantizar la durabilidad a largo plazo.

En caso de desastre, se restauran los datos desde las copias de seguridad en S3.

Base de Datos de Cache (ElastiCache - Redis)



La **Base de Datos de Cache (ElastiCache - Redis)** se utiliza para mejorar el rendimiento del sistema, proporcionando almacenamiento en memoria para datos frecuentemente solicitados. Al usar cache, se reducen los tiempos de respuesta en las consultas y se descarga la carga del servidor principal de base de datos.

1. Almacenamiento en Memoria

Los datos que son solicitados con frecuencia, como configuraciones, sesiones de usuario o resultados de consultas, se almacenan en la cache para acelerar el acceso.

Redis, el motor utilizado para la cache, soporta operaciones rápidas y almacenamiento en memoria, lo que lo hace ideal para optimizar el rendimiento en sistemas con alto tráfico de solicitudes.

2. Gestión de Expiración de Datos

Los datos almacenados en la cache no son permanentes; se establecen políticas de expiración para asegurar que la información sea relevante y no ocupe espacio innecesario.

Los datos caducados son eliminados automáticamente, permitiendo un uso eficiente del almacenamiento en memoria.

3. Copia de Seguridad de Cache

Para mejorar la resiliencia del sistema, se implementa una copia de seguridad automática de la cache, que permite restaurar datos en caso de un fallo o pérdida de memoria.

Aunque la cache es temporal, la persistencia de datos importantes puede ser crucial en algunas operaciones.

4. Interacción con Otros Servicios

El **API Gateway** accede a la cache para obtener datos rápidos que ya han sido almacenados previamente, lo que minimiza la necesidad de realizar consultas a la base de datos principal.

El **Servicio Core Bancario** y el **Servicio de Notificaciones** también almacenan y consultan datos en la cache para mejorar la eficiencia.

El **Servicio de Monitoreo y Alerta** supervisa el rendimiento de la cache, incluyendo el uso de memoria y la frecuencia de acceso a los datos.

5. Seguridad y Cifrado de Datos:

Los datos en ElastiCache están cifrados en tránsito mediante TLS, asegurando la protección de las comunicaciones con la cache.

Para proteger los datos en memoria, se habilita el cifrado en reposo en los nodos de Redis.

Se configura la autenticación obligatoria con Redis AUTH para garantizar que solo los clientes autorizados puedan acceder a la cache.

6. Esquemas de Alta Disponibilidad y Failover:

ElastiCache utiliza grupos de replicación con nodos primarios y secundarios para proporcionar alta disponibilidad.

En caso de un fallo en el nodo primario, un nodo secundario se promueve automáticamente para mantener la continuidad del servicio.

La replicación asíncrona garantiza que los datos se mantengan sincronizados entre los nodos.

7. Procesos de Backup y Recuperación ante Desastres:

Se programan backups automáticos diarios de la cache para capturar el estado de los datos en memoria.

Los backups se almacenan en Amazon S3, permitiendo la restauración en caso de que todos los nodos fallaran.

En situaciones de desastre, se puede restaurar desde los backups en S3 para reestablecer la cache con los datos más recientes disponibles.

3. Justificación de Tecnologías y Patrones

Tecnologías Elegidas:

- Amazon Web Services (AWS): Se eligió AWS por su capacidad de escalabilidad, alta disponibilidad y amplia gama de servicios administrados que facilitan la implementación de soluciones empresariales seguras. La integración con servicios como RDS, DynamoDB y ElastiCache proporciona una base robusta para la gestión de datos.
- Angular/React para SPA y Flutter/React Native para Aplicación Móvil: Estas tecnologías permiten la creación de aplicaciones de alto rendimiento y multiplataforma, asegurando una experiencia de usuario fluida.
- Amazon RDS (Base de Datos Relacional): Proporciona una solución administrada para bases de datos relacionales con opciones de réplica y recuperación ante desastres.
- Amazon DynamoDB (Base de Datos NoSQL): Utilizado para la auditoría, donde se necesitan consultas rápidas y una alta capacidad de escalabilidad horizontal.
- Amazon ElastiCache (Redis): Ayuda a mejorar el rendimiento de la aplicación mediante la cache en memoria para datos de acceso frecuente.

Patrones de Arquitectura Utilizados:

- Arquitectura de Microservicios: Permite la división del sistema en servicios independientes que pueden escalarse y gestionarse por separado.
- API Gateway: Actúa como un punto central de entrada para las solicitudes y proporciona un control de acceso unificado.
- Patrón Circuit Breaker: Implementado en la comunicación entre microservicios para prevenir fallos en cascada.
- Patrón de Cache de Datos: Utilizado para mejorar el rendimiento al almacenar en caché datos frecuentemente solicitados.

4. Seguridad y Autenticación

Flujo de Autenticación:

- OAuth 2.0: Se implementa para la autenticación de usuarios, permitiendo un flujo seguro con el uso de tokens de acceso. Se integra con AWS Cognito para la gestión de usuarios y soporte de autenticación multifactor.

- Autenticación MFA (Multifactor): Añade una capa extra de seguridad al requerir un segundo factor, como un código enviado por SMS o una aplicación de autenticación.

Medidas de Seguridad Implementadas:

- Cifrado en Tránsito y en Reposo: Todos los datos transferidos se cifran utilizando TLS, y las bases de datos tienen cifrado en reposo habilitado.
- Content Security Policy (CSP): Políticas que ayudan a prevenir ataques de inyección de scripts (XSS).
- Autenticación y Autorización Basadas en Roles (RBAC): Los servicios tienen permisos específicos, asegurando que solo los usuarios y servicios autorizados puedan acceder a los datos sensibles.
- Auditoría y Monitoreo de Seguridad: Los eventos de acceso y cambios críticos se registran en una base de datos de auditoría, lo que permite rastrear incidentes y cumplir con las normativas.

5. Alta Disponibilidad y Recuperación ante Desastres

Estrategias de Alta Disponibilidad:

- Réplica Multi-Zona: Servicios clave y bases de datos están replicados en múltiples zonas de disponibilidad para evitar interrupciones.
- Failover Automático: Servicios como RDS y ElastiCache están configurados para realizar un failover automático en caso de fallo en la instancia principal.
- Autoescalado de Servicios: Se utilizan políticas de autoescalado en los servicios backend para gestionar la demanda variable.

Estrategias de Recuperación ante Desastres:

- Backups Automatizados y Gestión de Versiones:

Se configuran copias de seguridad automatizadas para todos los servicios críticos, con una frecuencia diaria, semanal y mensual. Estas copias se almacenan en Amazon S3 utilizando cifrado para asegurar la protección de los datos.

Las copias de seguridad se replican en múltiples regiones para garantizar la disponibilidad de los datos en caso de un fallo regional.

Se implementa un sistema de gestión de versiones que permite restaurar copias específicas de acuerdo con la fecha o evento, optimizando la recuperación en incidentes específicos.

- Plan Detallado de Recuperación ante Desastres:

El plan de recuperación define pasos claros para restaurar servicios críticos y bases de datos en caso de fallo catastrófico. Esto incluye la recuperación de servidores, servicios en contenedores, y la infraestructura de red.

Se establecen procedimientos para la conmutación por error (failover) automática a instancias de respaldo en caso de detección de fallos graves, con conmutación manual como respaldo si el sistema automático falla.

Incluye roles y responsabilidades asignados al personal, así como canales de comunicación establecidos para la coordinación en caso de emergencia.

- Recuperación Puntual (Point-in-Time Recovery) para Bases de Datos:

Las bases de datos soportan la recuperación puntual, lo que permite restaurarlas a un estado específico (con una precisión de minutos) en caso de pérdida de datos, corrupción o errores humanos.

Este enfoque minimiza la pérdida de datos al proporcionar opciones de restauración granular, reduciendo el impacto en las operaciones.

- Replica de Lectura y Conmutación por Error (Failover):

Las bases de datos críticas están configuradas con réplicas de lectura en zonas de disponibilidad diferentes para garantizar la continuidad de operaciones.

En caso de fallo, el tráfico se redirige automáticamente a la réplica de lectura, que se puede promover a base de datos principal con muy poca latencia.

Las réplicas se sincronizan en tiempo real para asegurar que los datos estén siempre actualizados.

- Infraestructura Multi-Región y Escalabilidad:

Los servicios críticos se despliegan en múltiples regiones de AWS para asegurar la continuidad del servicio incluso ante fallos catastróficos en una región específica.

La arquitectura multi-región permite el balanceo de cargas y la distribución de solicitudes, lo que facilita la escalabilidad horizontal y la rápida recuperación en caso de fallos regionales.

- Automatización con Infraestructura como Código (IaC):

La infraestructura se gestiona con herramientas como Terraform y AWS CloudFormation, permitiendo la reinstalación de servicios y la recuperación rápida en nuevas regiones.

Esto garantiza que la infraestructura restaurada sea idéntica a la original, reduciendo errores de configuración.

- Pruebas Regulares de Recuperación ante Desastres:

Se realizan simulaciones de recuperación ante desastres de manera trimestral para probar la efectividad del plan de DR y actualizar procedimientos según sea necesario.

Las pruebas incluyen la recuperación de servicios críticos y la validación de la integridad de los datos restaurados, así como ejercicios de comutación por error.

- Monitoreo Proactivo y Alertas Automáticas:

Los sistemas de monitoreo (Amazon CloudWatch, Prometheus) detectan fallos potenciales en la infraestructura y emiten alertas automáticas a los equipos de soporte.

Las alertas se envían a través de servicios como Amazon SNS, integrándose con herramientas de notificación como Slack o Microsoft Teams para asegurar una respuesta rápida.

6. Integración con Servicios Externos y Gestión de Datos

Mecanismos de Integración:

API Gateway como Punto de Entrada Unificado: Gestiona la integración con servicios externos a través de API RESTful.

SNS/SES para Notificaciones Externas: Utilizados para enviar notificaciones por correo electrónico o mensajes SMS a los usuarios.

Integración de Pagos y Verificación de Identidad: El sistema puede conectarse con pasarelas de pago y servicios de verificación de identidad (por ejemplo, reconocimiento facial) a través de APIs externas.

Gestión de Datos:

Persistencia en Bases de Datos Adaptadas: Datos relacionales en RDS, datos no estructurados en DynamoDB, y datos en caché en ElastiCache para diferentes necesidades de almacenamiento.

Cifrado y Control de Acceso a Datos: Implementación de políticas para asegurar la integridad y privacidad de los datos almacenados.

7. Consideraciones Normativas

- PCI DSS para Manejo de Información Financiera: Se aplican las mejores prácticas de seguridad para la protección de datos de tarjetas de crédito.

- Ley de Protección de Datos Personales (GDPR): El sistema asegura que los datos del cliente se manejen de acuerdo con las regulaciones de privacidad.
- Conozca a su Cliente (KYC): Implementación de verificaciones de identidad y almacenamiento seguro de los datos del cliente.

8. Manejo de Costos

La arquitectura propuesta se ha diseñado teniendo en cuenta la optimización de costos, utilizando servicios de AWS que permiten un uso eficiente de los recursos y la implementación de prácticas para minimizar los gastos sin comprometer el rendimiento ni la seguridad. A continuación, se detallan algunas estrategias clave implementadas para el manejo de costos:

1. Uso de Instancias Reservadas y Spot Instances:

Para los servicios que no requieren alta disponibilidad constante, como tareas de procesamiento batch o servicios secundarios, se recomienda el uso de Spot Instances, las cuales ofrecen precios considerablemente más bajos en comparación con las instancias bajo demanda.

En el caso de servicios críticos y persistentes, se pueden usar Instancias Reservadas, que permiten un ahorro significativo cuando se compromete el uso por uno o tres años.

2. Escalabilidad Automática:

La configuración de Auto Scaling para servicios en Amazon ECS y Amazon EKS asegura que solo se utilicen los recursos necesarios en momentos de alta demanda. Esto evita costos adicionales por capacidad ociosa.

La escalabilidad automática de las bases de datos en Amazon RDS permite ajustar el tamaño de la instancia según la carga, optimizando el costo en función del uso.

3. Almacenamiento Optimizado:

Los datos infrecuentemente accedidos se almacenan en Amazon S3 utilizando clases de almacenamiento más económicas, como S3 Infrequent Access o S3 Glacier, reduciendo costos de almacenamiento sin perder acceso a datos históricos o de respaldo.

Amazon ElastiCache se usa para reducir la carga de las bases de datos y mejorar el rendimiento del sistema, evitando la necesidad de adquirir instancias de base de datos más grandes y costosas.

4. Optimización de Bases de Datos:

El uso de Amazon RDS para bases de datos relacionales y DynamoDB para almacenamiento NoSQL permite optimizar los costos de almacenamiento y recuperación de datos según el tipo de carga.

Las copias de seguridad automatizadas y la configuración de snapshots periódicos permiten un enfoque de almacenamiento eficiente, evitando costos elevados por almacenamiento de larga duración.

5. Uso de Servicios Gestionados:

Los servicios gestionados como Amazon Cognito para autenticación y Amazon SNS/SES para notificaciones reducen los costos de operación y mantenimiento, ya que no requieren la administración manual de la infraestructura subyacente.

6. Monitoreo de Costos y Alertas:

Se implementan prácticas de monitoreo de costos con AWS Cost Explorer y AWS Budgets, configurando alertas para mantener el gasto bajo control. Esto permite ajustar el uso de servicios en tiempo real para evitar sobrecostos.

7. Optimización de Tráfico y Balanceo de Carga:

El uso de Amazon API Gateway y Elastic Load Balancer (ELB) garantiza que el tráfico sea distribuido de manera eficiente, evitando costos elevados en picos de demanda.

Estas estrategias garantizan un equilibrio entre la eficiencia de costos y el rendimiento del sistema, asegurando que la arquitectura sea sostenible y escalable en el tiempo.

9. Conclusiones

La arquitectura propuesta cumple con los requisitos del sistema de banca por internet, permitiendo a los usuarios acceder a su historial de movimientos, realizar transferencias y pagos de manera segura. El diseño incluye medidas específicas para cumplir con normativas bancarias y estándares de seguridad, como el cifrado de datos, autenticación multifactor (MFA) y almacenamiento seguro de credenciales, garantizando así la protección de la información del usuario.

La solución presenta una segmentación clara de responsabilidades entre los diferentes servicios y componentes, lo que facilita el mantenimiento, la escalabilidad y el desarrollo de nuevas funcionalidades. La arquitectura orientada a microservicios permite el desacoplamiento de componentes, asegurando que cualquier cambio o actualización en un servicio no afecte a los demás, mejorando la resiliencia del sistema.

Se han elegido tecnologías de vanguardia como AWS, Angular/React, Amazon RDS, DynamoDB, entre otras, proporcionando una base sólida y escalable para el sistema. Los patrones de arquitectura utilizados, como la infraestructura multi-región, el uso de cache distribuido y el diseño basado en eventos para auditoría y monitoreo, aseguran una alta disponibilidad, baja latencia y tolerancia a fallos.

La implementación de OAuth 2.0 con AWS Cognito para la autenticación asegura un flujo seguro y eficiente de acceso para los usuarios, con soporte para MFA y recuperación de cuentas. Además, los servicios están protegidos contra ataques comunes mediante medidas de seguridad como Content Security Policy (CSP), protección contra XSS y CSRF, así como la encriptación de datos en reposo y en tránsito.

La solución implementa mecanismos de failover, replicación de datos y recuperación ante desastres para minimizar el tiempo de inactividad en caso de fallos. La infraestructura en la nube de AWS, con despliegue multi-región y automatización mediante IaC, garantiza que los servicios críticos permanezcan disponibles en todo momento.

La integración de servicios de cache como Amazon ElastiCache mejora el tiempo de respuesta para datos frecuentemente solicitados, reduciendo la carga sobre las bases de datos. Además, el monitoreo activo y el sistema de alertas automáticas aseguran que cualquier problema sea detectado y resuelto rápidamente, mejorando la excelencia operativa.

La arquitectura desacoplada y modular permite añadir nuevas funcionalidades y servicios con facilidad, adaptándose al crecimiento de la entidad BP y a las futuras necesidades del sistema. El uso de servicios gestionados en la nube facilita la escalabilidad horizontal, permitiendo ajustar la capacidad de los recursos según la demanda sin comprometer la calidad del servicio.

Finalmente, la adopción de microservicios y la separación clara de responsabilidades permiten un desarrollo ágil, con la posibilidad de realizar actualizaciones o despliegues sin afectar al sistema completo. La infraestructura como código (IaC) facilita la replicación del entorno para pruebas, actualizaciones y despliegues en nuevos entornos, reduciendo el riesgo de errores humanos.

En resumen, la solución de arquitectura para el sistema de banca por internet no solo cumple con los requisitos técnicos y normativos, sino que también proporciona una base sólida y escalable para el futuro crecimiento de la entidad BP, con un enfoque en la seguridad, la alta disponibilidad y la excelencia operativa.