Desafío Técnico Devsu - Banca por Internet

Indice

Indice de Figuras

Banca por Internet

En el dinámico y competitivo mundo de la banca, la arquitectura de soluciones de un sistema bancario juega un papel crucial para asegurar la eficiencia operativa, la seguridad y la innovación continua. La arquitectura de soluciones se refiere al diseño y organización de los componentes tecnológicos y funcionales que permiten el funcionamiento integral de los servicios bancarios.

En este contexto el siguiente ejercicio explica como los diferentes componetes interactuan entre si para satisfacer los requerimientos tecnicos solicitados por la empresa BP a travez de diseño de arquitectura C4 con las vistas de Contexto, Contenedores y Componentes

Contexto de la Solución

El Sistema de Banca Digital existen muchos componentes indispensables interrelacionados en donde detallamos algunos importantes para su funcionamiento como:

- Sistema de Autenticación.
- Sistema para manejo de la ley de Protección de datos de los usuarios.
- Sistema de Notificaciones.
- Sistema de Logs.
- Sistema de Detección de Fraudes.
- Sistema de Producto de Datos.
- Sistemas Core.

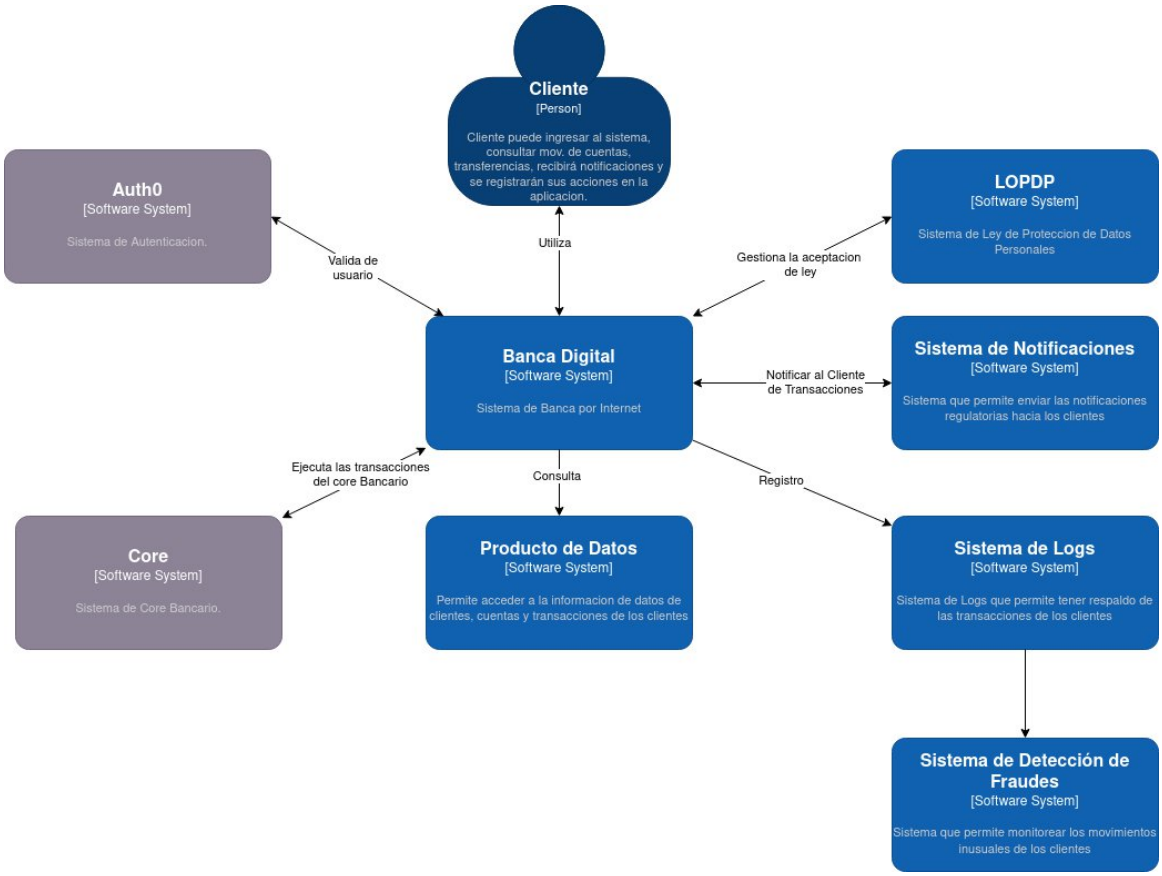

Figure 1 – Diagrama Contexto de Banca Digital

Detalle de la Solución

Sistema Banca Digital

En este ejercicio de arquitectura se ha tomado en cuenta dos aplicaciones para acceso a la Banca por Internet, una para acceso por web y otra aplicación móbil, se sugiere las siguientes tecnologías:

- AngularJs, Reactjs para el front de banca web dado que el framework de angular es bastante maduro y estable en sus librerias, asi mismo como alternativa se sugiere Reactjs dado que esta libreria es utilizada ampliamente y tiene gran soporte de la comunidad.
- Tiene un punto de acceso que es el ApiGateway que proporciona seguridad, monitoreo, y facilita a los desarrolladores la creacion, mantenimieto, publicacion de la APIs.
- Tiene una capa de microservicios en donde separamos las responsabilidades de integracion con los servicios legados comunmente que acceden a los Cores u otros sistemas antiguos de las empresas de manera que podamos poner una fachada

necesaria para estandarizar los contratos tecnicos de los servicios, y por otro lados los servicios de nuevas capacidades con desarrollo nativo de nube y que estan ligados a la logica de negocio del funcionamiento del sistema de banca digital.

- Esta integrado a un sistema de logs que mas adelante estaremos detallando en el presente documento y este a su vez alimenta al sistema de antifraudes que permite al banco detectar de manera temprana los posibles movimientos no autorizados por los clientes.
- Se integra ademas a un Sistema para manejo de la ley organica de proteccion de datos de los usuarios LODPD, importante para darle la facilidad al cliente de aceptar o nedegar en cualquier momento y a su ves el sistema de banca digital pueda determinar a quienes se pueden o no procesar los datos personales y ofrecer productos personalzados.
- Debe contar un Sistema robusto de notificaciones necesarios para el envio de mensajes al cliente, parte fundamental ya que es normativo que se envien como minimo 2 notificaciones por diferentes medios por cada transaccion monetaria.
- El sistema producto de datos permite habilitar la informacion de los clientes de manera rapida y oportuna en este caso especifico para poder escalar la alta demanda de las consulta de datos, aunque tambien elproducto de datos nos permite aprovechar los grandes volúmenes de datos, técnicas de análisis, y modelos de inteligencia artificial o aprendizaje automático para ofrecer soluciones o insights a los usuarios.

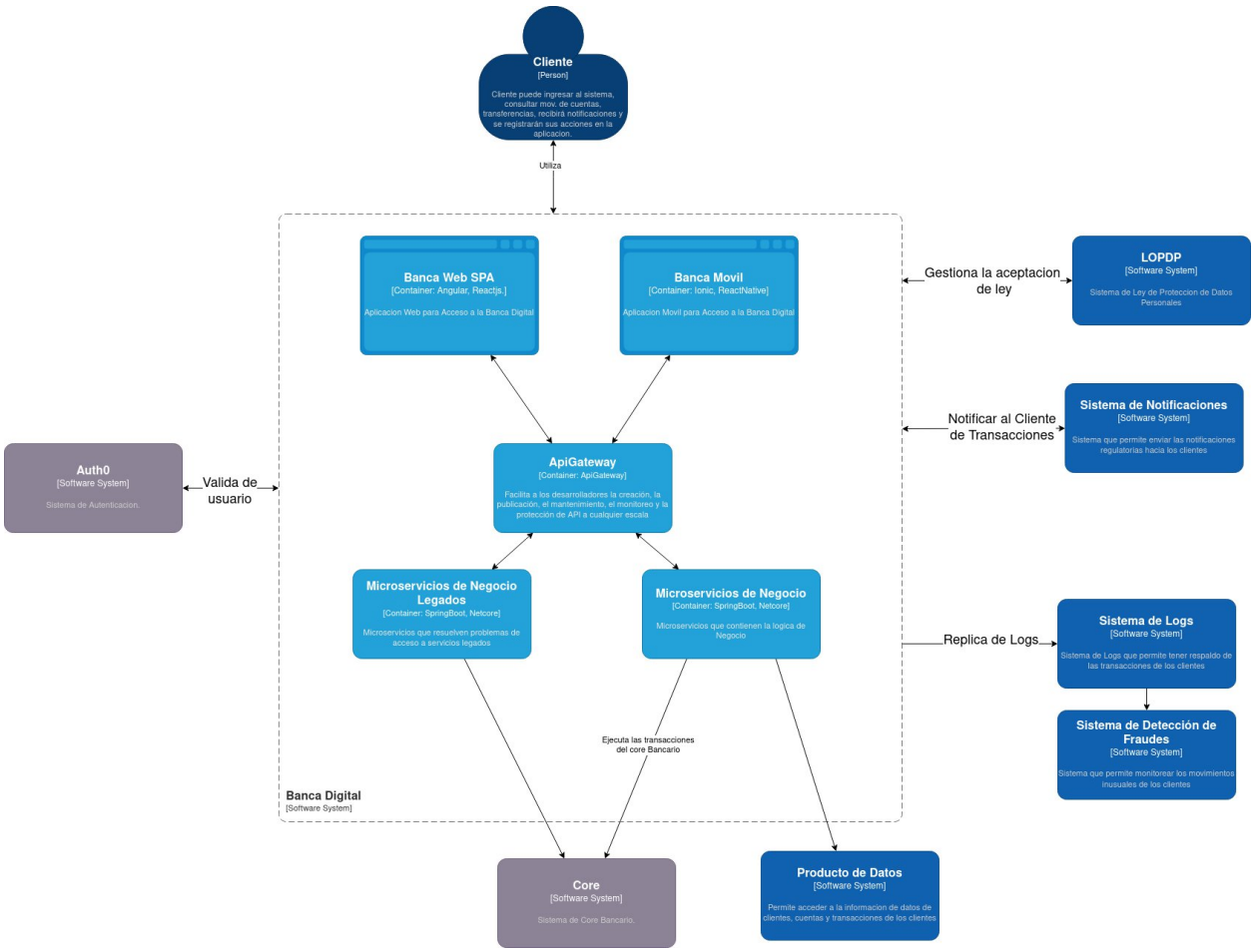

Figure 2 – Diagrama Contenedor Banca Digital

En el diagrama componentes de Banca Digital se establecen algunos principios de diseno para la excelencia operativa:

- La de las APIs se establecen a traves del Apigateway por medio de la autentucación de usuario que puede ser por usuario y contrasena, huella digital y pin. Una vez realizada la autenticación el acceso las APIs se validara con JWT.
- Se establecen en la capa nativa (web / hibrida) de front de las aplicaciones web y móbil, las funcionalidades propias de cada canal que por lo general son el acceso a la aplicación, vista consolidada.
- Se establece una capa común de front que puede ser expuesta a traves de webviews para simplificar el desarrollo de las capacidades y proporcionar una misma experiencia de usuario.
- Se establece un patron de diseno Backend for Frontend para la integración de las Apis y la capa de front con la responsabilidad de resolver las necesidades específicas de cada canal, como por ejemplo, personalización de gráficos, formateo de mensajes.

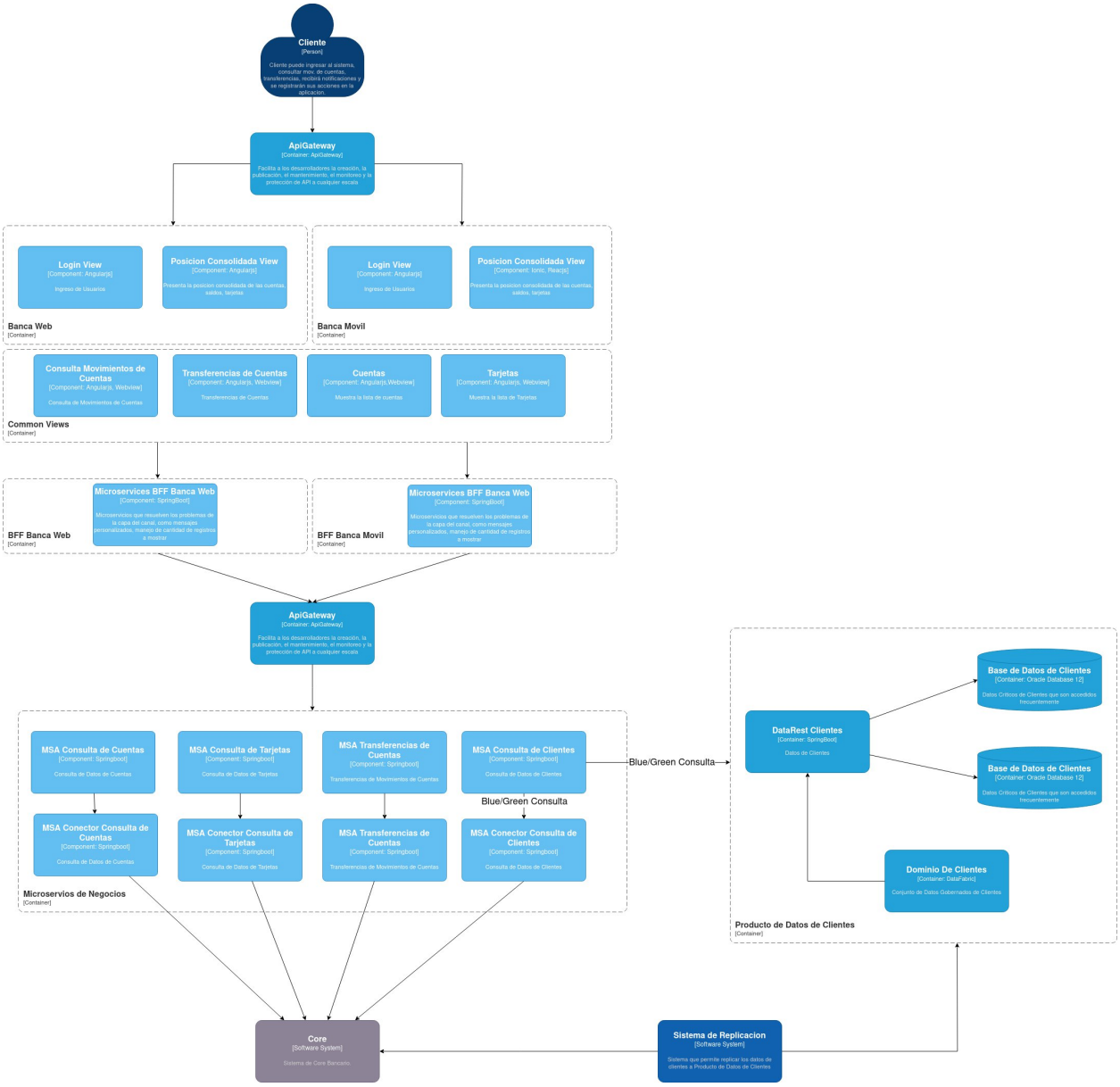

Figure 3 – Diagrama Componente de Banca Digital

- En la capa de microservicios backend se sugiere que se desarrolle en Springboot o .Net C# que son muy buenos leguajes que poseen buena intergacion para solucioes cloud native, donde se construiran a traves de la interfase RESTfull.
- Se separa la logica de negocio y la funcionaliudad de integracion de la capa subyacente (core, o cualquier otro sistema legado) con eso garantizamos la segregación de responsabilidades de los servicios.

Sistema de Autenticación

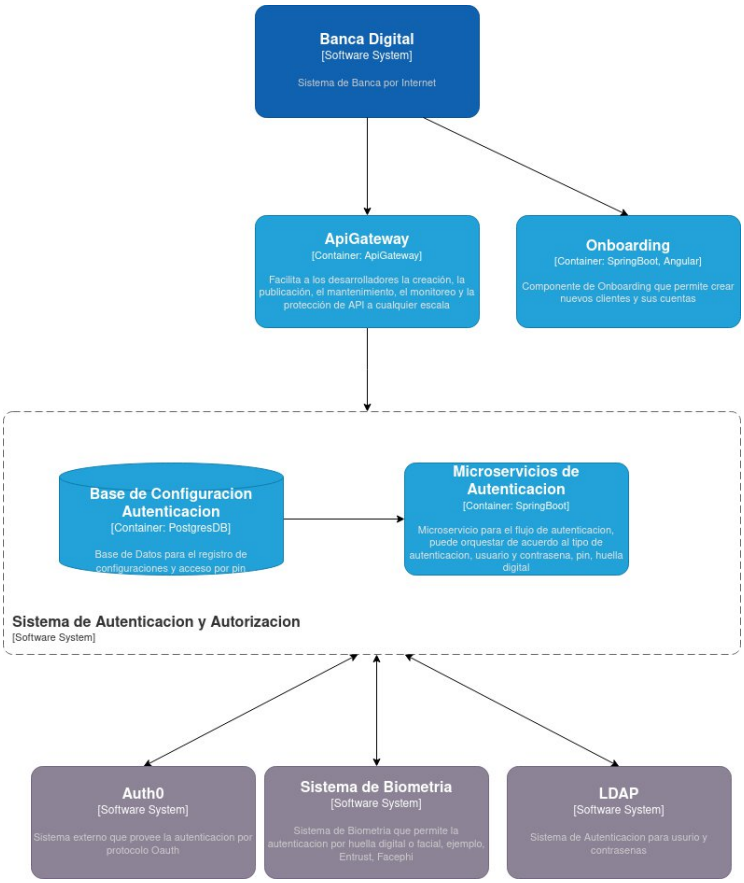

Figure 4 – Diagrama Contenedor Autenticación

Basicamente el Sistema de autenticacion comprende en los siguientes componentes:

- API de autenticación y autorización, diseñada para soportar la carga transaccional, y la integración con los sistemas externos de que proveen el acceso mediante usuario y contraseña, por pin y por huella dactilar.
- Sistemas externos de autenticación, Oauth, Biometria y LDAP, un punto principal a la hora de elegir cualquier sistema o herramienta de nube es que por regulación del ente de control SB, deben contar con las certificaciones ISO27001, ISO 27017 y ISO27018.
- Replicación de datos de clientes nuevos una vez creados en el Sistema Onboarding para que puedan enrolarse en la Banca Web o Móvil.

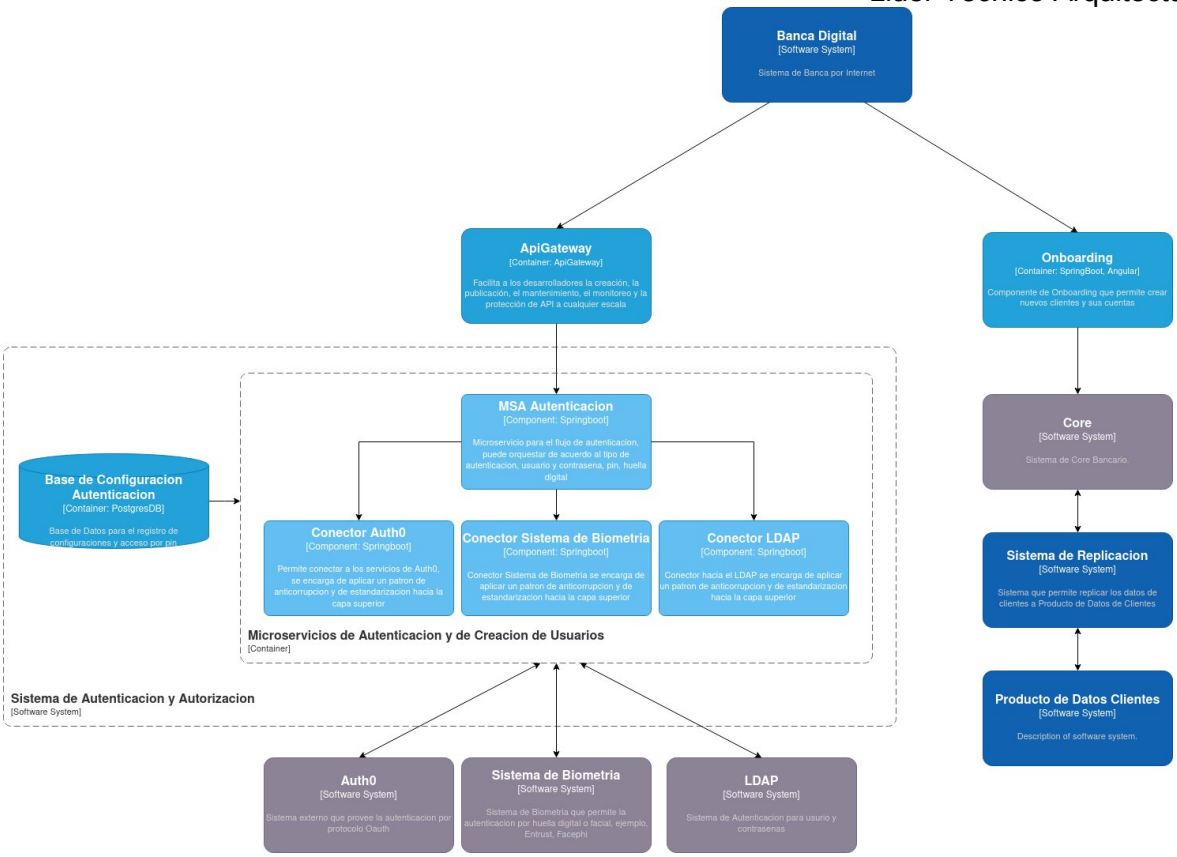

Figure 5 – Diagrama Componente Autenticacion

- Se separa la lógica de negocio de la orquestación de la autenticación con la integración de los servicios o plataformas de servicios externos a los cuales les hemos denominado microservicios conectores.
- Replicación de los datos al Producto de Datos de Clientes.
- El sistema onboarding deberá permitir la creación de los clientes con cedula, pasaporte, y el sistema de autenticación de la misma forma permitir el enrolamiento con estos tipos de identificación.

Sistema Notificaciones

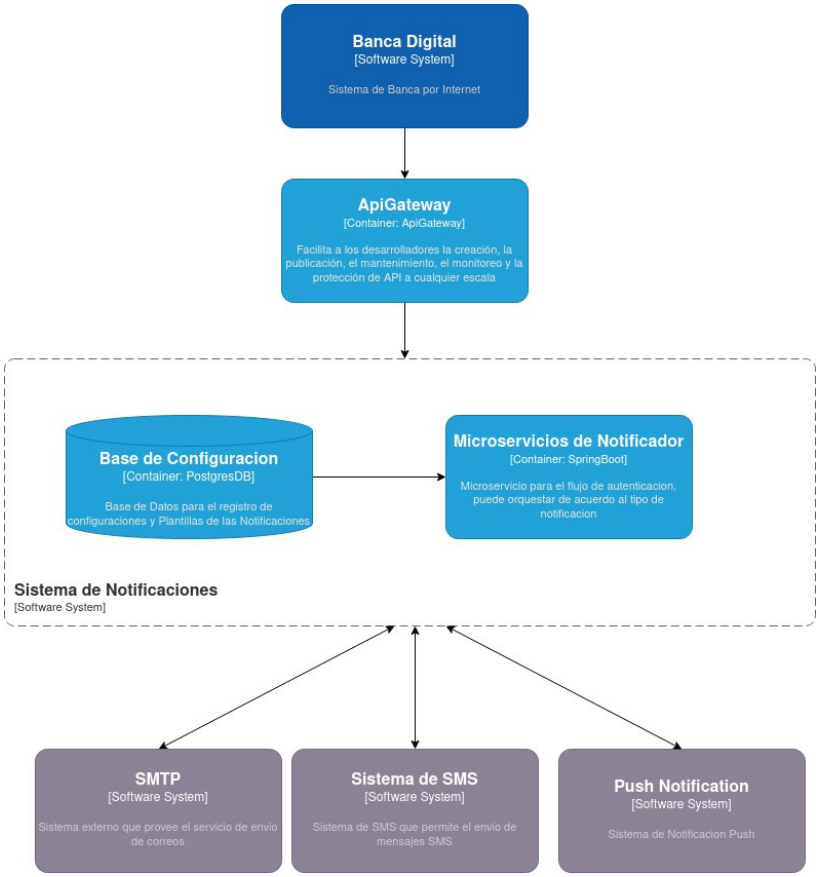

Figure 6 – Diagrama Contenedor Notificador

El Sistema de Notificaciones está compuesto de la capa de integración con la interfaz RESTfull, microservicios con la logica de negocio para reseolver las diferentes casuisticas de tipo de mensajes por canal y los Sistemas externos para el envio de los mensajes como SMTP, SMS y Push.

Todo sistema de notificacion deber tener los siguientes elementos importantes:

- Entrega de Notificaciones, multi-canal, fiabilidad y escabilidad
- Personalización, relevancia y preferencias de usuario
- Contenido de la notificacion, claridad del mensaje y podria tener un call to action.
- Gestion de Notificaciones, historial, manejo de estados, categorias
- Seguridad y Privacidad, bases de datis aseguradas, validar con la LOPDP, aunque este ultimo tiene que ver con mensajes de ofertas.
- Monitoreo y Analisis, metricas para el seguimiento y trazabilidad.

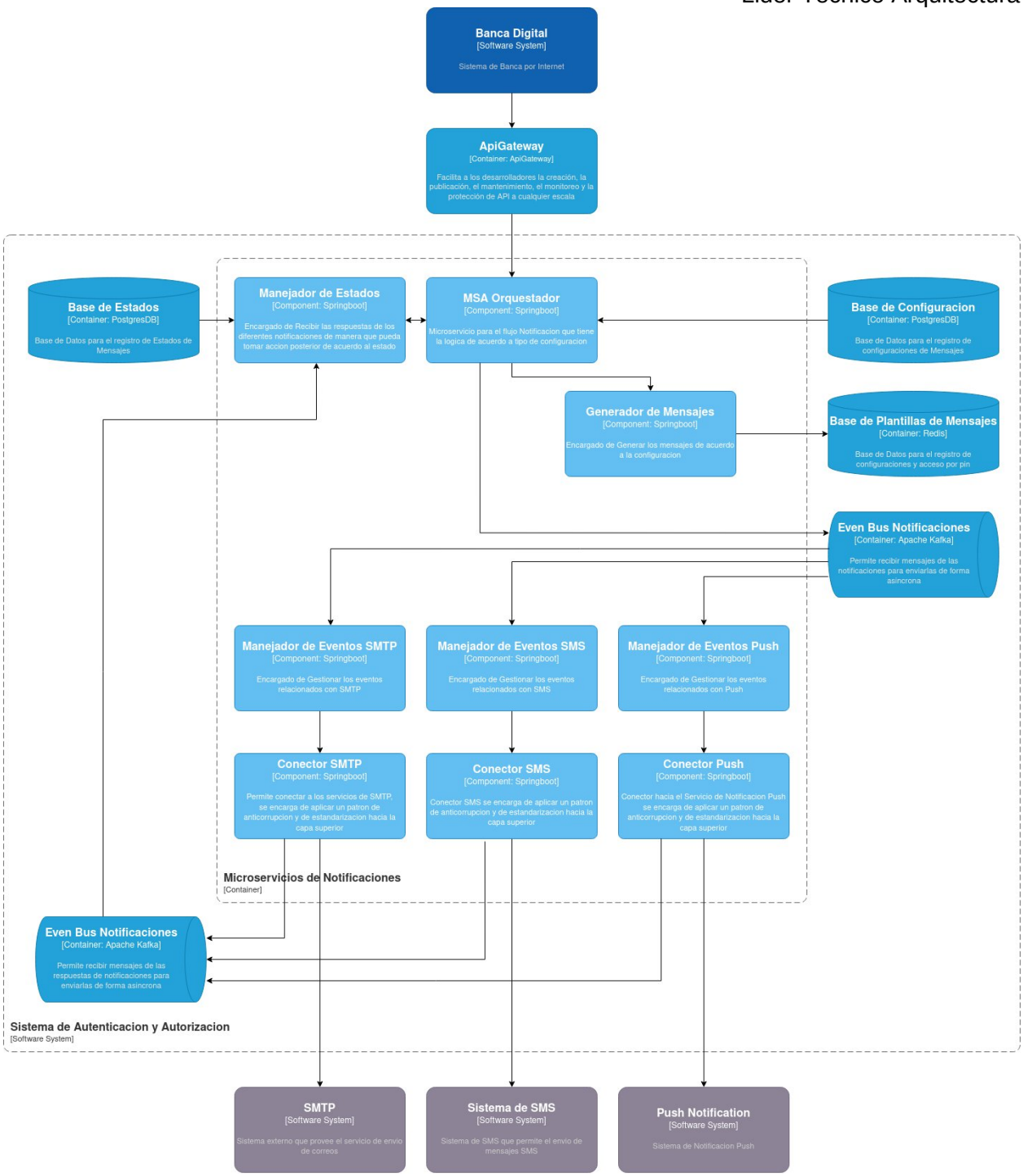

Figure 7 – Diagrama Componente de Notificador

El funcioamiento del sistema de notificaciones se comprende de la siguiente forma:

- Exposicion del API a través de APIGateway
- Microservicio de Orquestacion de envio de mensajes que posee la logica de como y por que medio se debe enviar el mensaje
- Microservicio de Generacion de Mensaje que es capaz de poder crear los mensajes de acuerdo al canal, medio y aplicación.
- Microservicios de escucha para manejar los envios de los mensajes hacia los diferentes sistemas externos con sus microservicios conectores que resuelven la complejidad de integración.

- MIcroservicio de manejo de estados que permite obtener el resultado del envio y será capaz de poder resolver los reintentos en el caso de falla, del cual estará combinado con un dieseno de patron Saga.
- Un bus de eventos para el manejo de envio de mensaje de manera asincrona para soportar altos volumenes de transaccionalidad.

Sistema de Logs

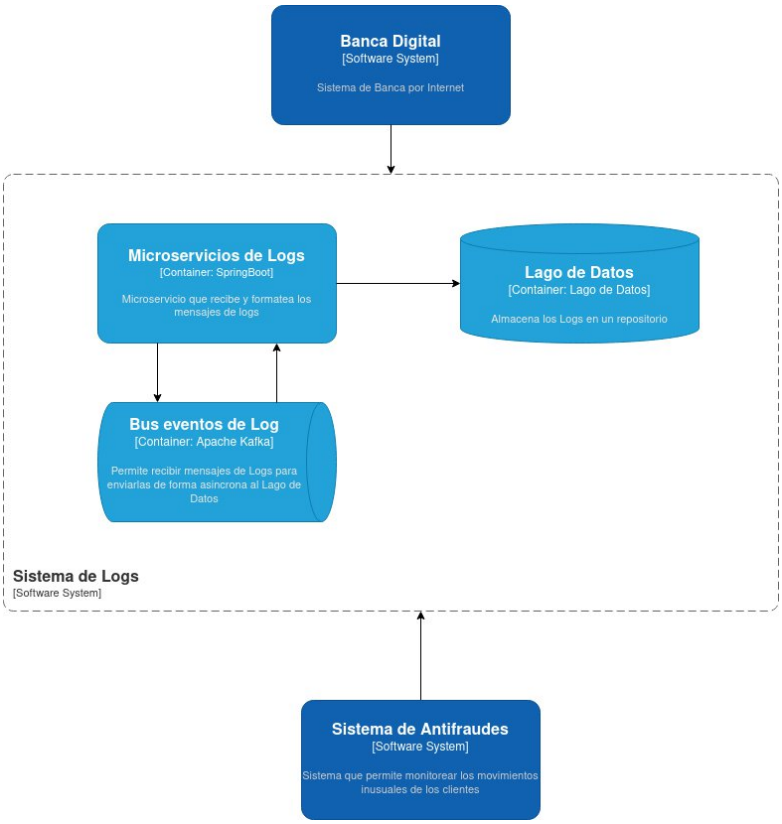

Figure 8 – Diagrama Contenedor de Sistema de Logs

El sistema de logs, es una parte clave en un sistema de Banca DIgital y como no decir para cualquier sistema, no permite el monitoreo, diagnostico, auditoria y mantenimiento de aplicaciones, de igual forma permite alimentar a otros sistemas como por ejemplo sistema de deteccion de fraudes.

Para que este sistema sea efectivo debe realizar lo siguiente:

- Capura de Eventos de microservios y otros componentes que conforman la solución
- Almacenamiento de logs, que comunmente es llevado hacia el lago de datos, para su posterior formateo, aplanado y analisis de los datos.
- Niveles de Logs, que dependiendo de la aplicación podemos dar profundidad del tipo de mensaje y su utilidad
- Resilencia y Disponiblidad, que son importantes para que todo sistema aseguras el correcto funcionamiento.

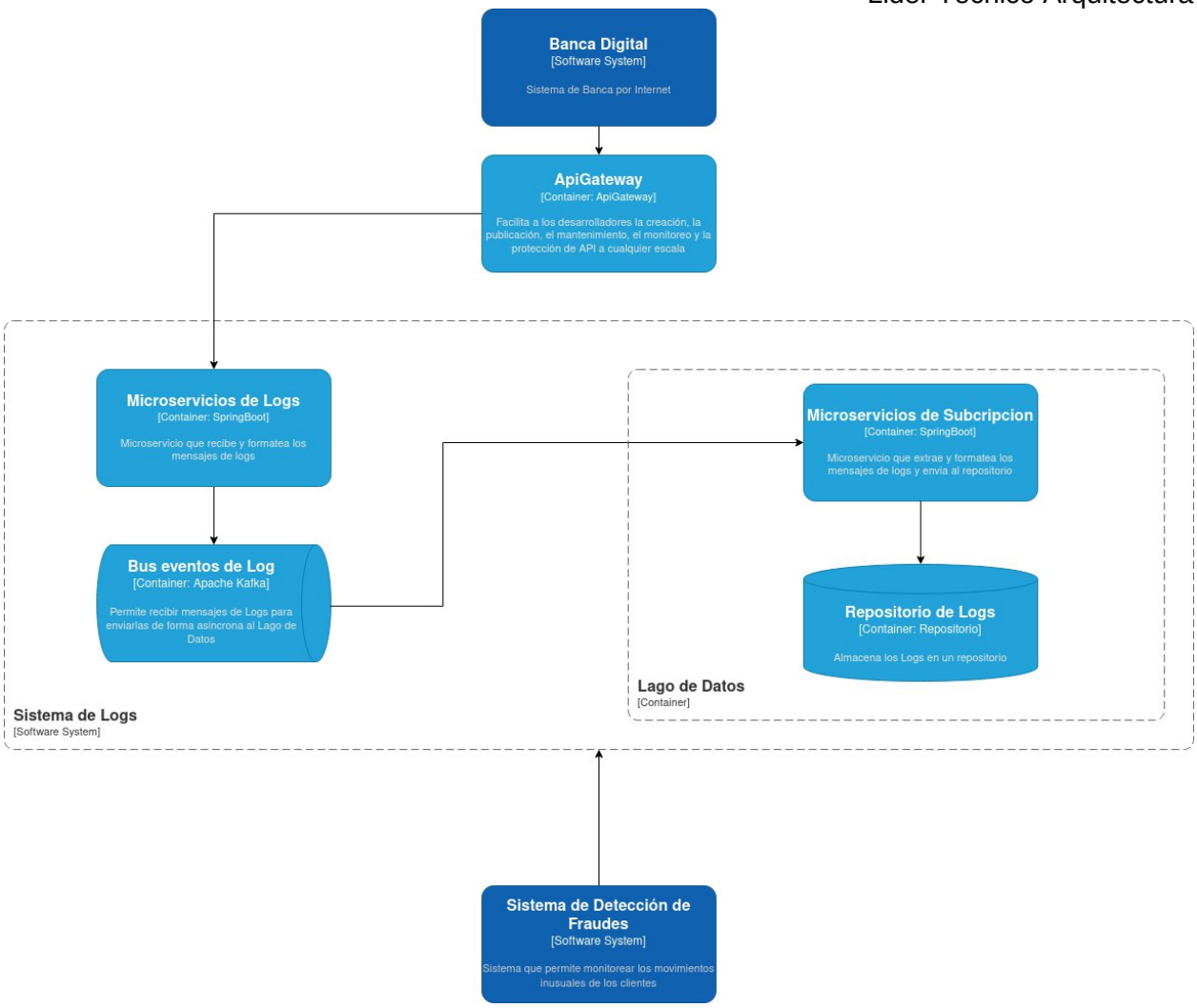

Figure 9 – Diagrama Componente Sistema de Logs

Aca vemos los siguientes componentes dentro de la solución:

- Un API expuesto para el envio de los mensajes, se propone que sea websocket para establecer una conexion rapida.
- Los microservicios que manejen los diferentes formatos de recepcion de mensajes
- Un bus de eventos para soportar alta carga transaccional
- Microservicios de suscripcion hacia el bus de eventos y posterior almacenamiento al lago de datos
- El repositorio o lago de datos para poder almacenar a largo plazo, recordemos que los mensajes de auditoria deben tener por lo menos 7 años de almacenamientos y en otros casos pordia llegar a 10 años.
- Otra estrategia para la mejor adopción de logs e integración en el desarrollo es la creacion de una libreria en el caso de Springboot es con log4j o slf4j a traves de un appeder que este congurado en con el Api y que sea importado dentro de las librerias del proyecto de desarrollo y configurado en los properties del mismo.

Infraestructura Cloud - Arquitectura de Referencia

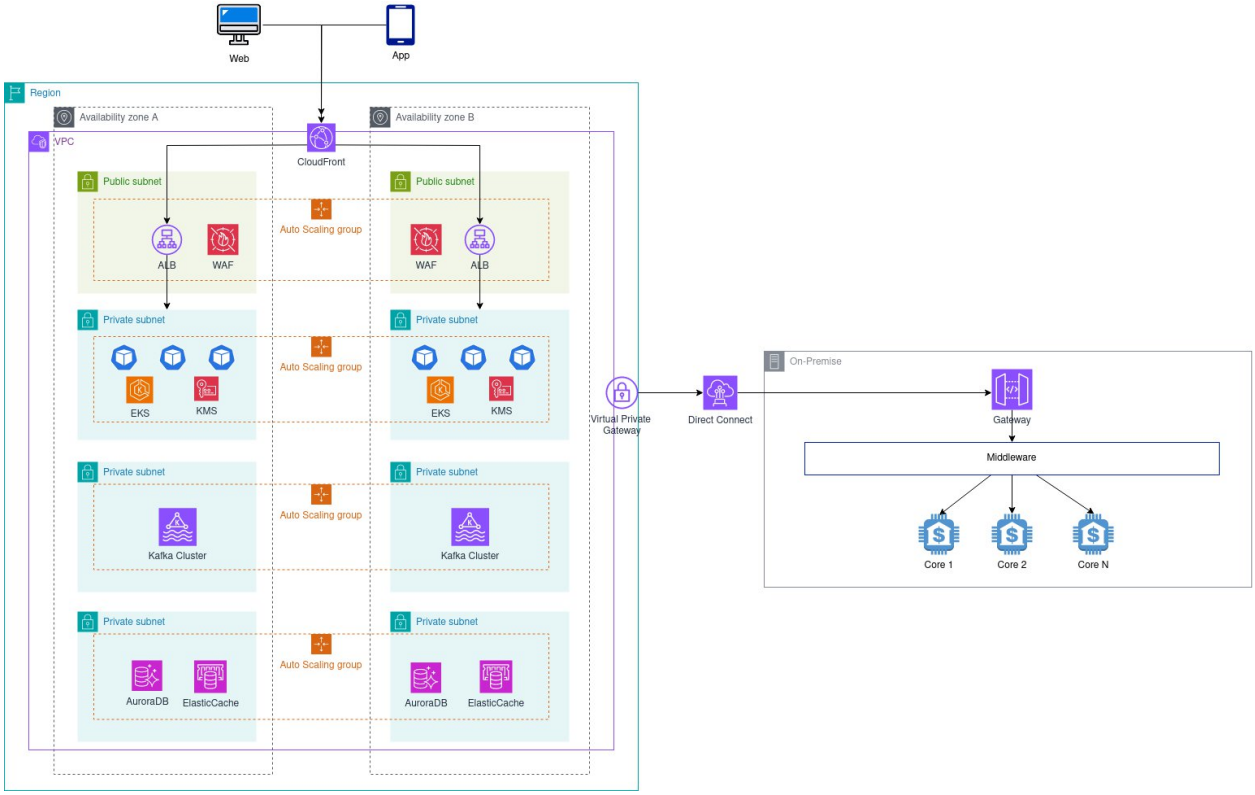

Figure 10 – Diagrama de Infraestructura Cloud

En este Diagrama de Arquitectura de Referencia en AWS podemos encontrar algunos aspectos que han sido considerados para garantizar alta disponibilidad, tolerancia a fallos, recuperaciónante desastres.

- Se habilitan una region y dos zonas de disponibilidad donde la recuperacion podria ser en minutos si se requeriere que sea en tiempo real deberiamos configura poner dos regiones con la misma configuración.
- Una subnet publica para la configuracion de ALG (Aplication Load Balancer), WAF y Apigateway
- Una subnet privada para la capa middleware, microservicios y keystore management para almacenamiento de los secretos, esto con la finalida de que los microservicios sean configurados de manera segura. Los microservicios estan montados sobre EKS en modo Cluster en las dos zonas de disponibilidad.
- Una subnet privada para la capa del Bus de Eventos que se sugiere que se utilice MSK de AWS en cluster en las dos zonas de disponibilidad
- Una subnet privada para la capa de almacenamiento de datos, AuroraDB, ElasticCache, S3
- Configuraciones de seguridad que van a depender de los accesos requeridos por cada uno de los comoponentes ubicados en cada subnet privada.
- Conexion a una Virtual Private Gateway con una configuracion de Direct Connector para el enlace al sistema core onpremisse.