

# Diseño Arquitectónico

## “Banca Virtual”

### Contexto y alcance del sistema

El contexto de la solución se enmarca en un sistema de banca por internet, cuyo alcance se delimita en la posibilidad de acceder al histórico de movimientos, realizar transferencias y pagos entre cuentas propias e interbancarias de los clientes.

### Metas y restricciones arquitectónicas

Toda la información referente al cliente se tomará de 2 sistemas:

- Una plataforma Core que contiene información básica de cliente, movimientos, productos y,
- Un sistema independiente que complementa la información del cliente cuando los datos se requieren en detalle.

El sistema utilizará sistemas externos o propios de envío de notificaciones, mínimo 2. Para lo cual se propone la utilización de servicio institucional de Correo Electrónico, así como la contratación de un servicio para el envío de mensajes SMS a celulares.

El sistema contará con 2 aplicaciones en el Front:

- Una aplicación SPA y,
- Una aplicación móvil desarrollada en un Framework multiplataforma.

Ambas aplicaciones autenticarán a los usuarios mediante un servicio que usa el estándar OAuth2.0, para el cual no requiere implementar toda la lógica, ya que la compañía cuenta con un producto que puede ser configurado para este fin.

Tenga en cuenta que el sistema de Onboarding para nuevos clientes en la aplicación móvil usa reconocimiento facial, por tanto, su arquitectura deberá considerarlo como parte del flujo de autorización

y autenticación, a partir del Onboarding el nuevo usuario podrá ingresar al sistema mediante usuario y clave, huella o algún otro método.

El sistema utiliza una base de datos de auditoría que registra todas las acciones del cliente y cuenta con un mecanismo de persistencia de información para clientes frecuentes, para complementar y conseguir el objetivo deseado, se propone la utilización del patrón de diseño Event Sourcing.

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

## Consideraciones

Para este reto, mencione aquellos elementos normativos que considere importantes tener en cuenta para entidades financieras, Ejemplo ley de datos personales.

Garantice alta disponibilidad (HA) y tolerancia a fallos, (DR), Seguridad y Monitoreo.

Si lo considera necesario, su arquitectura puede contener elementos de infraestructura en nube, Azure u AWS, garantice baja latencia, cuenta con presupuesto para esto.

El modelo debe ser desarrollado bajo modelo c4 (Diagrama de Contexto, Contenedores y Componentes), describa hasta el modelo de componentes.

## Arquitectura

Para el envío de notificaciones se propone la utilización del servicio institucional de Correo Electrónico, ya que el mismo cumple con las necesidades del negocio y se evita la inversión en una opción diferente. Se propone también, la contratación de un servicio para el envío de mensajes SMS a celulares, de esta manera se abarca a la mayoría de los modelos de teléfonos celulares que puedan utilizar los clientes. Para ambos servicios se tendrá especial cuidado en el cumplimiento de la Ley Orgánica de Protección de Datos Personales (LOPDP).

Para el desarrollo de la solución de Banca Virtual se propone la utilización de tecnología Microsoft específicamente la última versión de .NET con el lenguaje de programación C#, misma que tiene soporte multiplataforma y permitirá desarrollar con un solo IDE y un solo lenguaje de programación el Sitio Web SPA como la aplicación para dispositivos móviles; tampoco es necesario realizar una inversión ya que el IDE Visual Studio dispone de una versión Community, misma que es muy completa y permite a los desarrolladores reducir el tiempo en las tareas de codificación.

En el Front End se propone 2 opciones:

1. Para el Sitio Web SPA, se propone la utilización de HTML5+CSS+Blazor (tecnología que permite la ejecución de código C# en el lado del Navegador Web), y para la aplicación móvil se propone el

uso de MAUI (tecnología multiplataforma para el desarrollo de aplicaciones móviles con .NET con C#), de esta manera no es necesario disponer de personal con conocimientos diferentes a .NET y C#.

2. Para el Sitio Web SPA, se propone la utilización de HTML5+CSS+Vue.js, y para la aplicación móvil se propone el uso de IONIC.

Para el proceso de autenticación y autorización, se dispone de un servicio que implementa OAuth2.0, por lo que no es necesario su desarrollo, más se deberá configurar el servicio para su utilización con la aplicación de la Banca Virtual, tomando en cuenta que, la aplicación móvil, podrá utilizar autenticación biométrica, para lo cual se deberá seguir el siguiente flujo.

#### Sitio Web:

- Solicitar usuario y contraseña

#### Aplicación Móvil

- Solicitar usuario,
- Solicitar contraseña o,
- Solicitar Huella digital o,
- Solicitar reconocimiento facial

#### Sitio Web y Aplicación Móvil

- Obtener Token
- Obtener Permisos
- Iniciar sesión y aplicar permisos

En cuanto al registro de Auditoría, se propone la utilización del patrón de diseño Event Sourcing, para de esta manera poder disponer de información detallada e histórica por cada cambio que se realice al estado de los datos, así como la posibilidad de reconstruir información puntual a un estado anterior mediante la ejecución de los eventos pasados. Este patrón de diseño facilita el cumplimiento de los requisitos regulatorios al proporcionar un registro inmutable de todas las operaciones. Se tendrá especial cuidado en dar cumplimiento a lo establecido en la LOPDP.

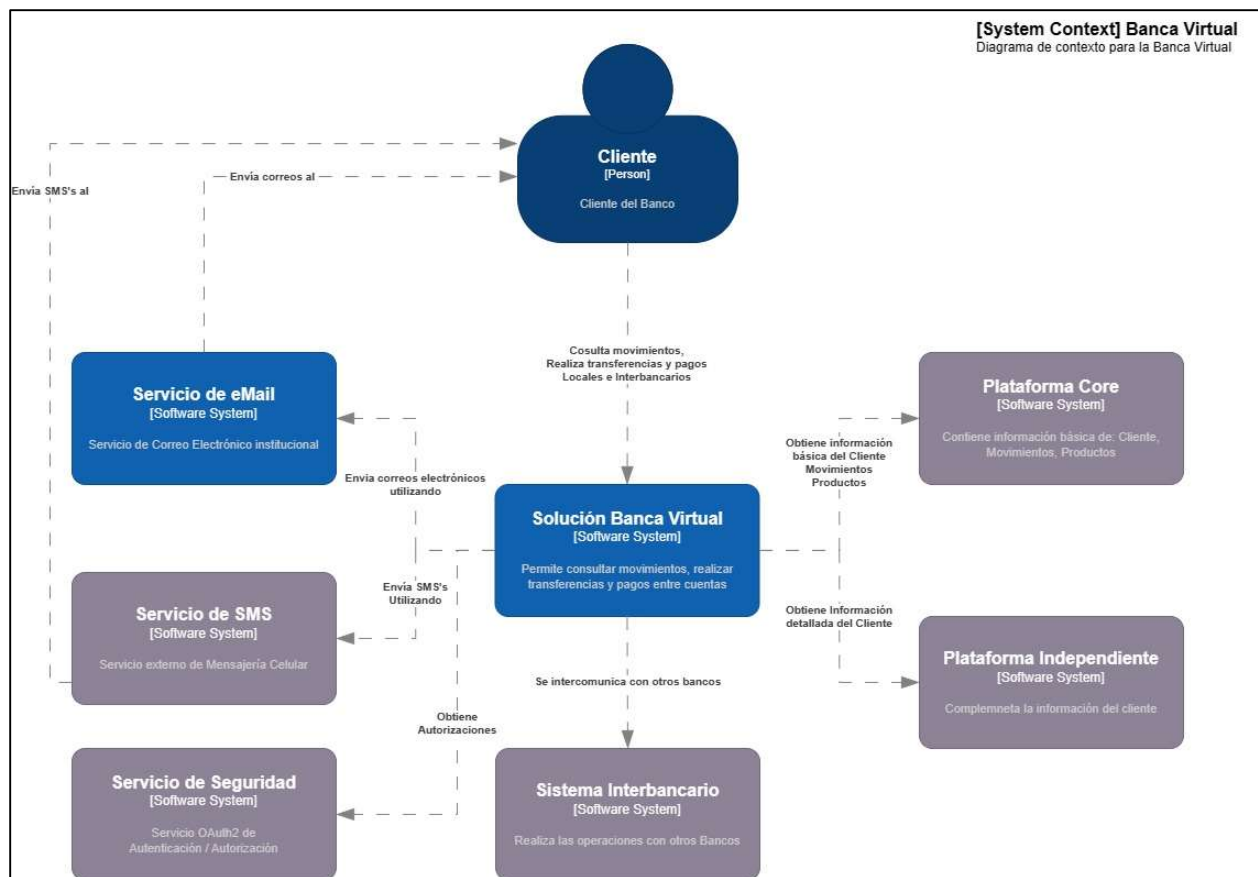
Para los Microservicios se propone adoptar:

- Domain Driven Design (DDD), para que estos se encuentren alineados con los dominios y lenguaje del negocio,
- Patrón Command Query Responsibility Segregation CQRS, para la optimización de la lectura de información y reducir posibilidades de ataque que pretendan modificar datos,
- Patrón SAGA para garantizar la atomicidad de la información.

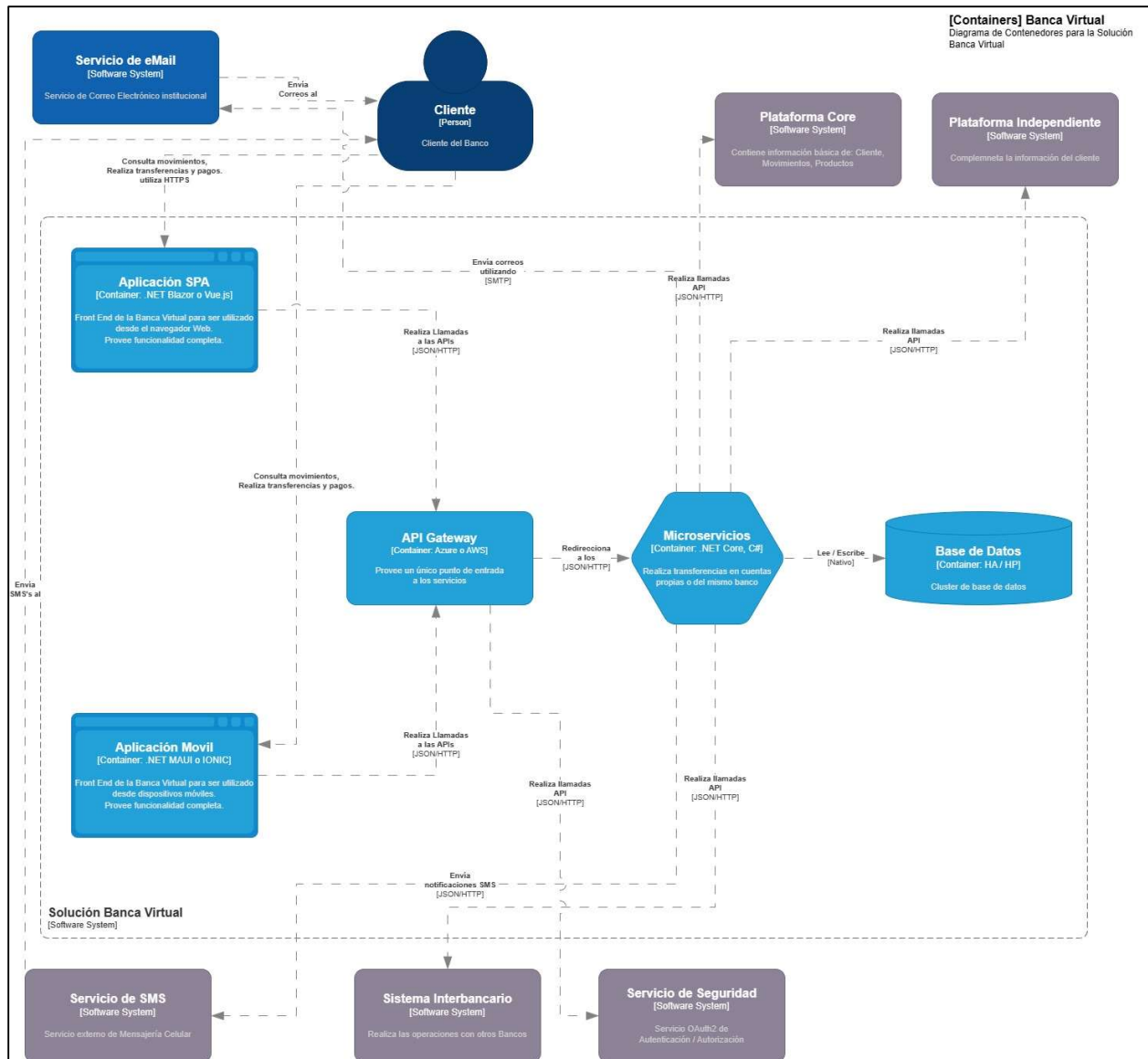
La funcionalidad se la implementará en librerías, mismas que para su despliegue, se lo puede hacer desde Azure Functions, AWS Lambda o desde API Web con Docker utilizando Azure AKS o Azure Container Apps o también AWS ECS o AWS Fargate.

Para el monitoreo de la aplicación se utilizará el conjunto completo de herramientas de Telemetría que dispone .NET para realizar el chequeo de salud y métricas, las cuales las podemos monitorear mediante Application Insights y Azure monitor o AWS CloudWatch, o también con las herramientas Prometheus y Grafana.

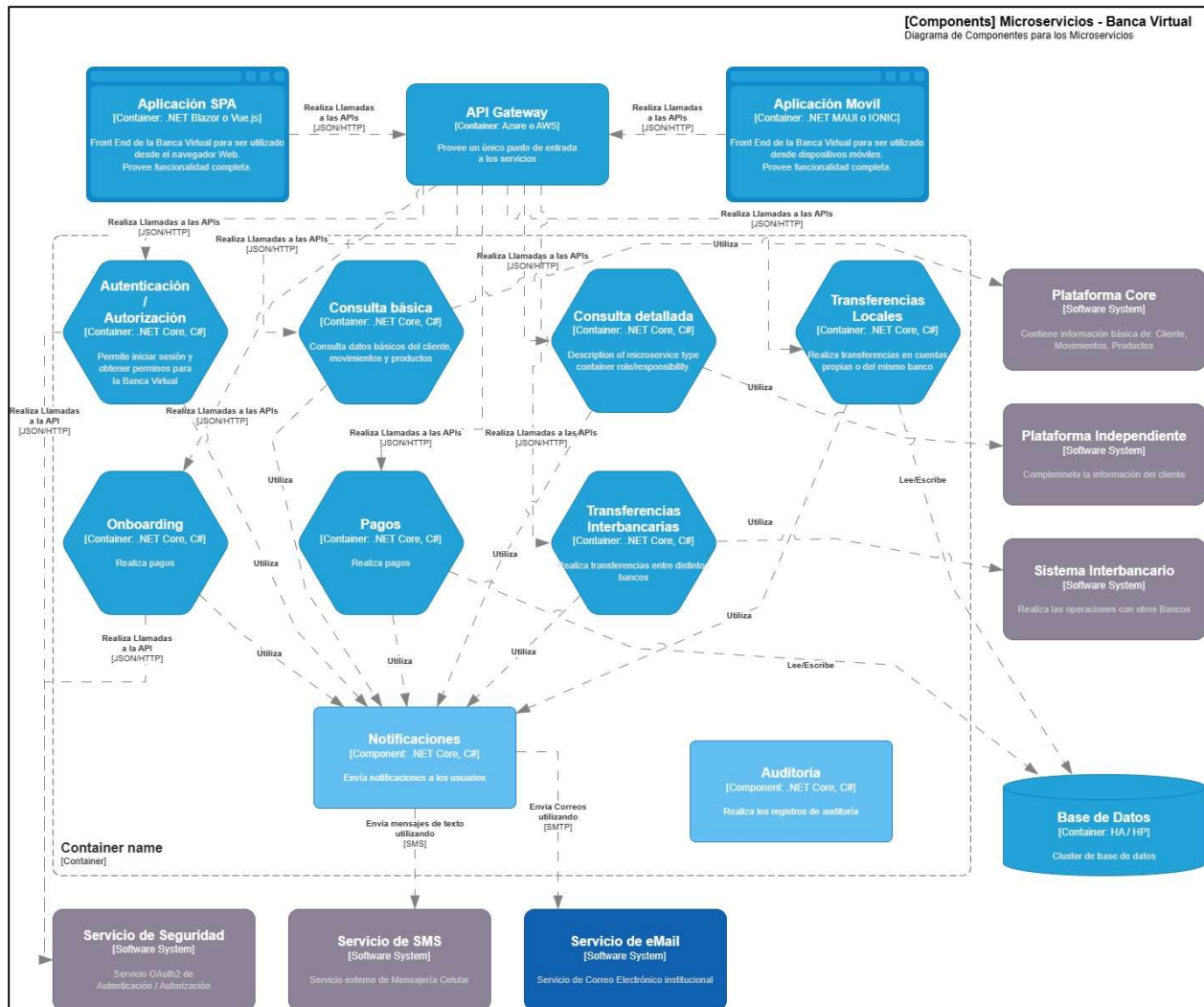
## Diagrama de Contexto



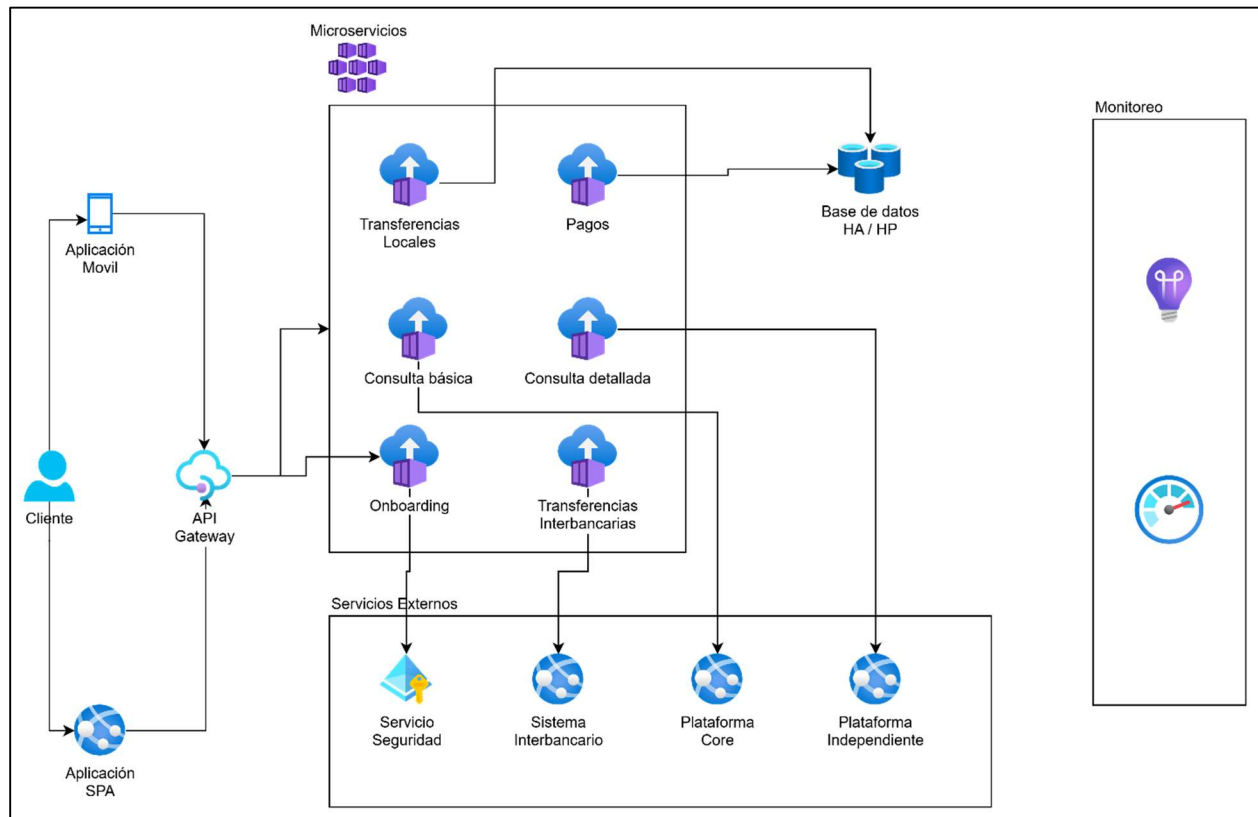
## Diagrama de Contenedores



## Diagrama de Componentes



## Diagrama de Infraestructura



Por simplicidad se provee únicamente el diagrama basado en la nube publica de Microsoft Azure.