

**BP BANCA EN LINEA**

## Contenido

Introducción .....	3
Cumplimiento de las normativas de seguridad.....	3
Alcance .....	4
Tecnologías propuestas.....	4
Frontend.....	4
Backend.....	4
AWS.....	5
Arquitectura del sistema.....	5
Componentes principales.....	5
Consideraciones generales.....	5
Modelo de contexto del sistema.....	6
Modelo de Contenedores .....	7
Modelos de Componentes .....	8
API Autenticación .....	8
API Onboarding .....	9
API Consulta Datos Básicos .....	10
API Consulta Movimientos .....	11
API Transferencias.....	12
API Auditor .....	13
API Notificaciones .....	14
SPA Cliente .....	15
APP Móvil .....	16
SPA Auditor .....	17
API Monitoreo .....	18
API Clientes Frecuentes.....	19
Conclusiones .....	19

## Introducción

En este documento se describe la arquitectura propuesta para el sistema de Banca en Línea para la entidad BP.

La Banca en Línea permitirá a los usuarios realizar operaciones financieras como consultas de movimientos, transferencias entre cuentas de la entidad e interbancarias, pagos a terceros y onboarding.

La solución se diseña para cumplir con los requisitos de tolerancia a fallos, recuperación ante desastres, seguridad y monitoreo, excelencia operativa y auto-healing.

## Cumplimiento de las normativas de seguridad

Para las entidades financieras es importante tener en consideración medidas de seguridad y aceptación de responsabilidades entre las que se encuentran, pero no está limitado a:

- Uso de MFA para la autenticación y funcionalidades que modifiquen la información sensible del usuario.
- Notificación al cliente por varios canales, de acciones ejecutadas sobre funcionalidades sensibles.
- Renovación periódica de las contraseñas de acceso de los usuarios.
- Registro de datos del usuario relevantes para la seguridad como direcciones IP o código del dispositivo al realizar transacciones.
- Cerrar sesión luego de un tiempo de inactividad.
- Manejo de roles.
- Políticas de buenas prácticas de nombre de usuario y contraseñas, y mantener un historial para que estas no se repitan.
- Los canales de comunicación deben estar cifrados.
- Implementar herramientas que detecten de manera oportuna comportamientos inusuales.
- Implementar técnicas de hardening para registrar los dispositivos utilizados por los usuarios.
- Establecer límites autorizados de transacciones.
- Realizar los movimientos en tiempo real.
- Bloquear las cuentas de los usuarios cuando se detecten comportamientos inusuales como cuando los intentos de acceso fallidos superen un máximo definido (tres intentos).
- Procedimientos para afiliar, cancelar, suspender y reactivar un usuario de la banca.
- Implementar medidas de seguridad como CORS, rate limiting y protección contra ataques comunes.
- Aceptación para el tratamiento de datos personales.

## Alcance

Este proyecto abarca el diseño de una solución que incluye:

- Aplicación web SPA y aplicación móvil.
- Integración con el core financiero de la entidad y otros servicios externos.
- Implementación de los mecanismos de autenticación y autorización de usuarios mediante credenciales, PIN o huella dactilar.
- Implementación del mecanismo de onboarding de clientes con reconocimiento facial.
- Consultas de cuentas y los movimientos de estas.
- Transferencias institucionales e interbancarias.
- Pagos a terceros
- Registrar las acciones de los usuarios.
- Notificaciones por correo electrónico y SMS.

## Tecnologías propuestas

### Frontend

- **Web:** React.
- **Móvil:**

**MAUI:** Forma parte del ecosistema de .NET lo que permite aprovechar el conocimiento actual de C# que también es utilizado en el backend. Está posicionado como el futuro del desarrollo multiplataforma. Integra controles de UI mejorados y proporciona una flexibilidad a la hora de crear experiencias similares a las nativas (Android y iOS). Implementa Hot-reload en tiempo de desarrollo lo que ayuda a mejorar los tiempos y el debug.

**React Native:** Utiliza React como framework base por lo que se puede integrar con la aplicación web. La curva de aprendizaje no es muy pronunciada.

### Backend

- **Framework:** ASP NET Core Web API.
- **Bases de datos:** MySQL, MongoDB.
- **Mensajería:** Redis.
- **Comparación de imágenes faciales:** CompreFace.
- **Librerías varias:** EF Core, Ocelot, Hangfire, HealthChecks, MailKit, Serilog, Swagger, Metrics, Mapster, Mediator, Otp.Net

## AWS

- **Contenedores:** Docker.
- **Orquestación y escalabilidad:** ECS.
- **Base de datos:** Aurora, DynamoDB
- **Monitoreo:** CloudWatch
- **Seguridad y redes:** ELB, VPC.

## Arquitectura del sistema

### Componentes principales

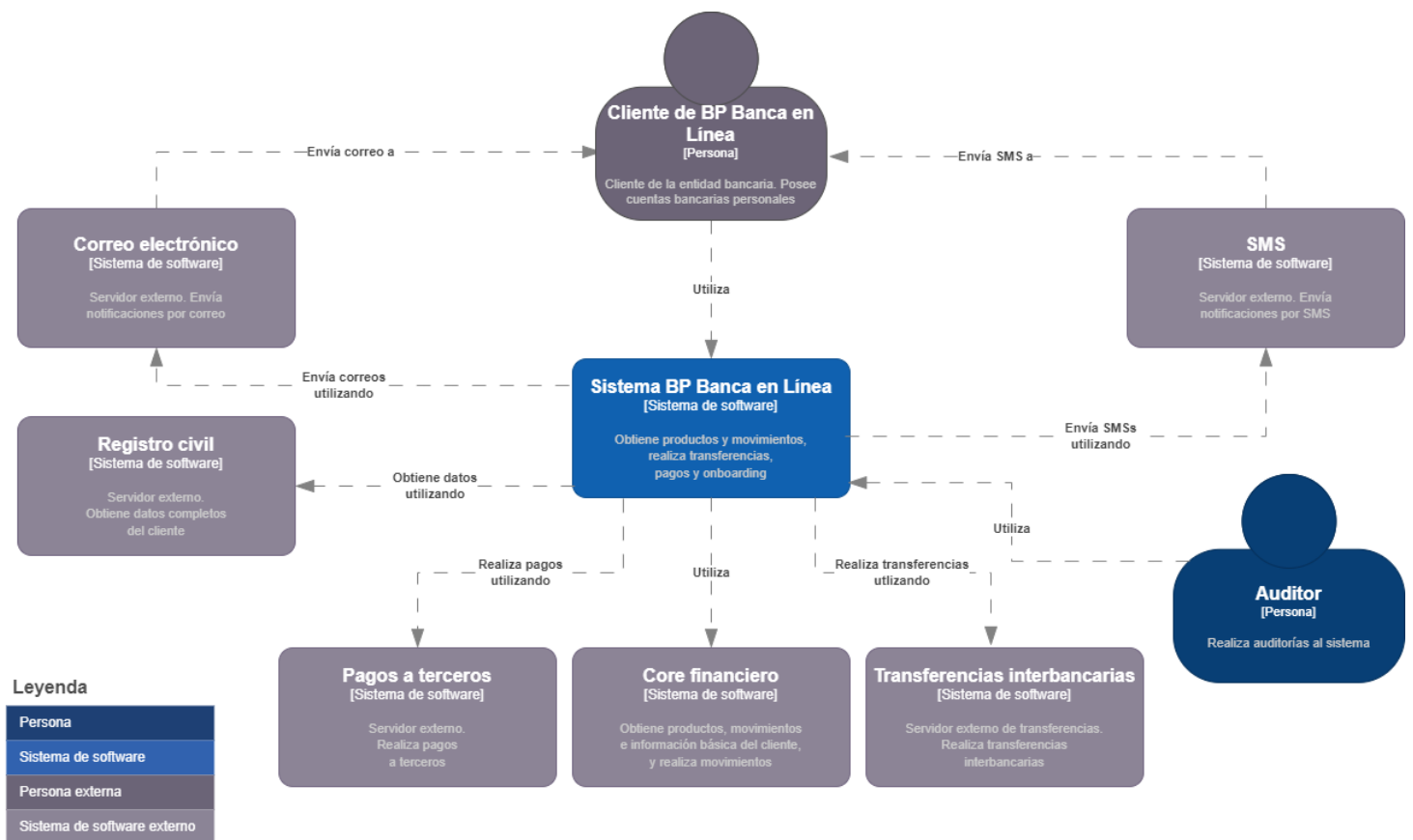
- **Frontend:** Aplicación móvil, aplicaciones web SPA Cliente y SPA Auditor.
- **Microservicios:**
  - **API Gateway:** Punto de entrada para todas las solicitudes, gestionando la comunicación entre el frontend y el backend.
  - **API Gateway Interno:** Permite al auditor del sistema obtener información de los servicios de auditoría y monitoreo.
  - **Autenticación:** Autentica al usuario.
  - **Onboarding:** Registra un nuevo cliente.
  - **Consulta Datos Básicos:** Devuelve los datos básicos del cliente y productos.
  - **Consulta Movimientos:** Devuelve los movimientos de un producto.
  - **Transferencias:** Realiza transferencias internas, interbancarias y pagos a terceros
  - **Notificaciones:** Envío de notificaciones usando servicios externos.
  - **Auditor:** Registra las acciones de los usuarios, errores y determina clientes frecuentes.
  - **Monitoreo:** Realiza el monitoreo de salud constante sobre todos los servicios internos y externos.
  - **CompreFace:** Compara imágenes faciales.
  - **Clientes Frecuentes:** Gestiona los clientes frecuentes del sistema.
- **Bases de Datos:**
  - **Bd Acciones:** Almacena las acciones de los usuarios, errores y clientes frecuentes.
  - **Bd Errores:** Almacena los errores del sistema.
  - **Bd Clientes Frecuentes:** Almacena los clientes frecuentes.
- **Bus de mensajes:** Direcciona los eventos del negocio entre componentes.

### Consideraciones generales

- El diseño del sistema cuenta con doce microservicios.
- La autenticación/autorización se realiza mediante el uso de tokens JWT generados a partir de un certificado RSA.

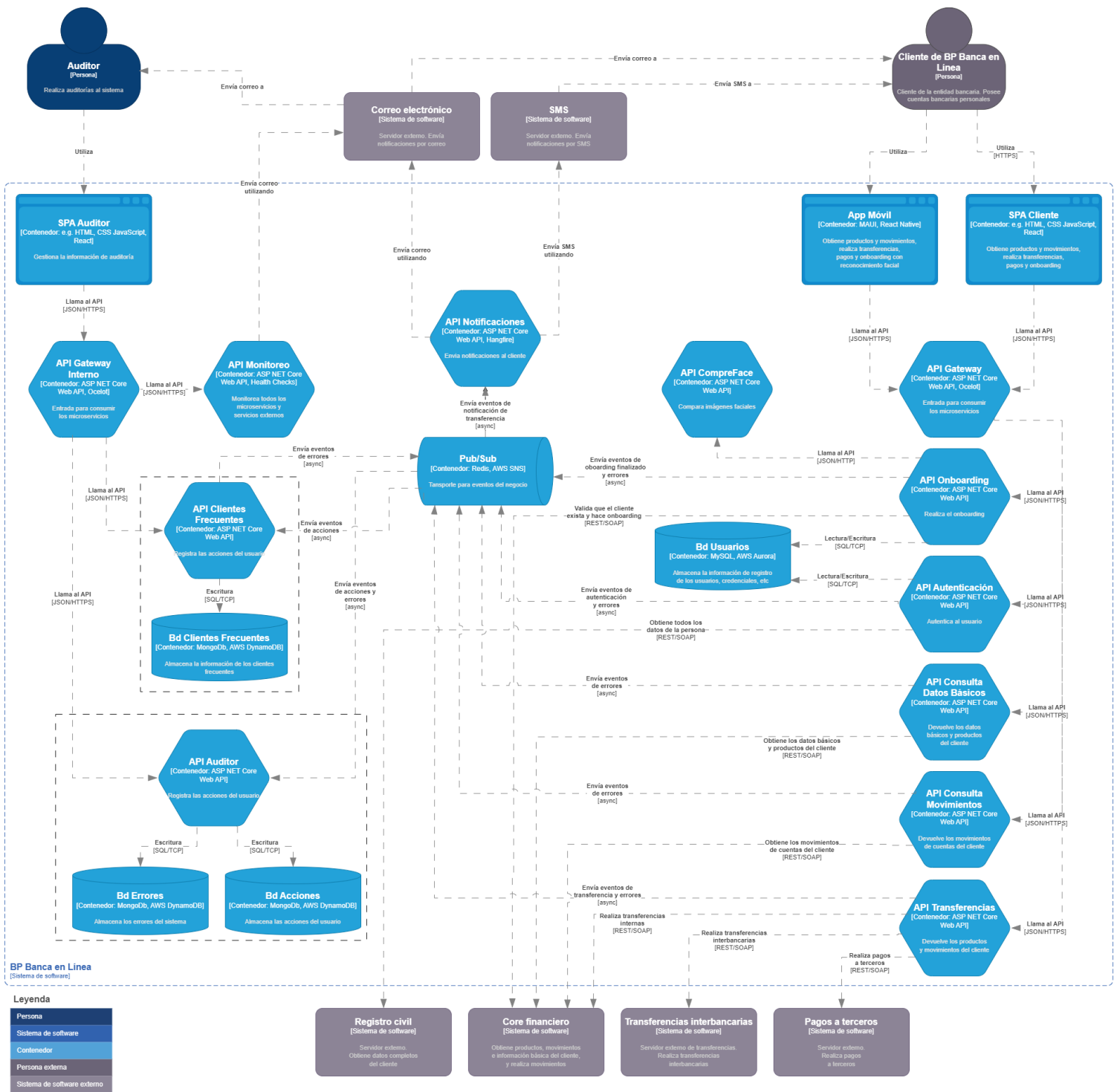
- Se utiliza MFA para la autenticación y transferencias haciendo uso de códigos OTP. Estos son generados continuamente en la aplicación móvil.
- Existe un servicio de monitoreo encargado de mantener una vigilancia continua sobre el estado de salud del resto de servicios internos y externos, y enviar notificaciones en caso de fallos.
- Existe un servicio encargado de comparar imágenes faciales.
- Se configuran dos canales para el envío de eventos que son, acciones del usuario y registro de errores.
- Se implementa una cola en el envío de notificaciones para que sean enviadas de manera asíncrona y con posibilidad de reintentos en caso de fallos.
- Desde la aplicación es posible autenticar mediante credenciales, desde la aplicación móvil es posible autenticar mediante credenciales, huella o PIN.

## Modelo de contexto del sistema



[Contexto del sistema] Sistema BP Banca en Línea

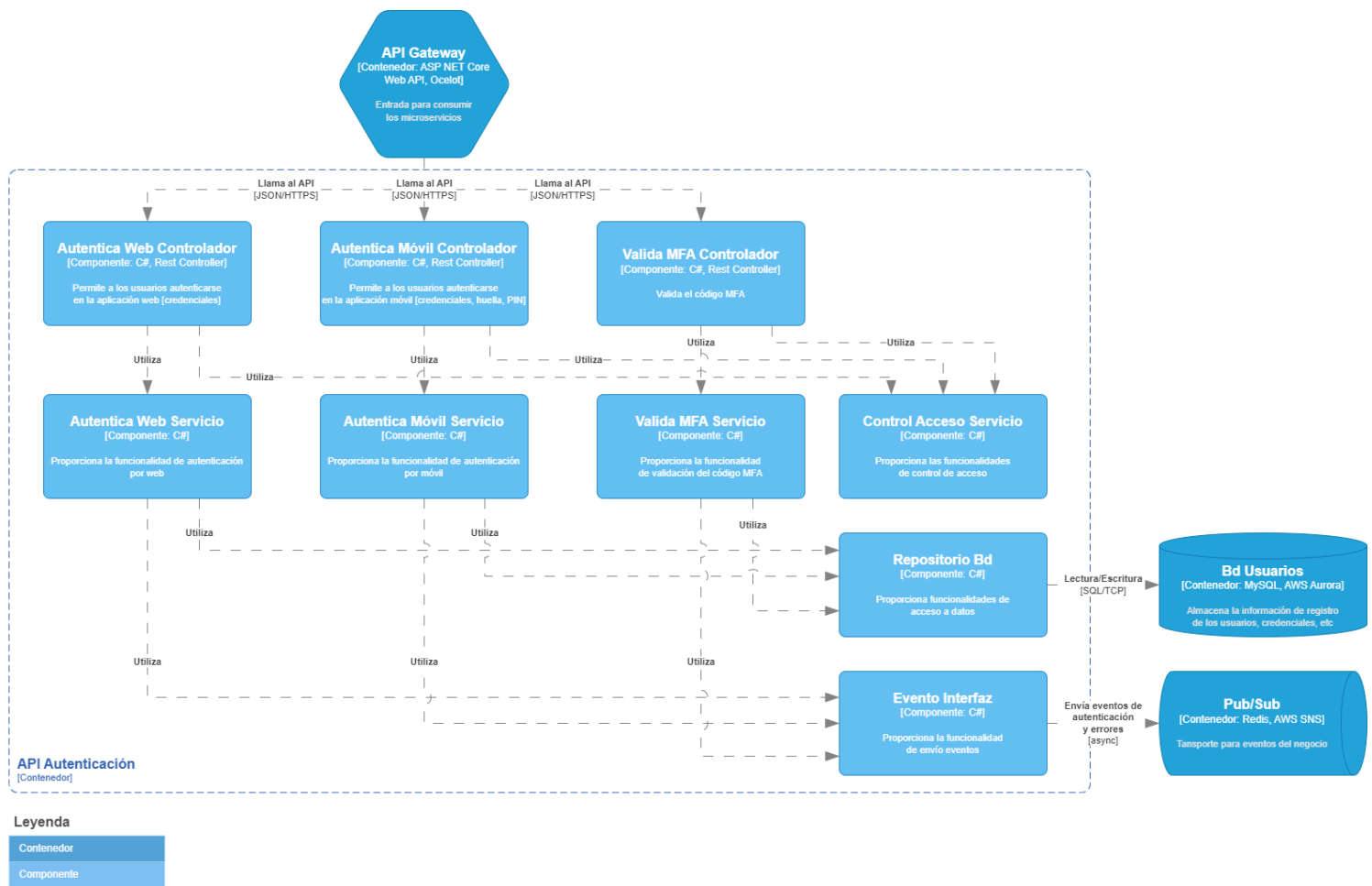
# Modelo de Contenedores



[Contenedor] BP Banca en Línea

# Modelos de Componentes

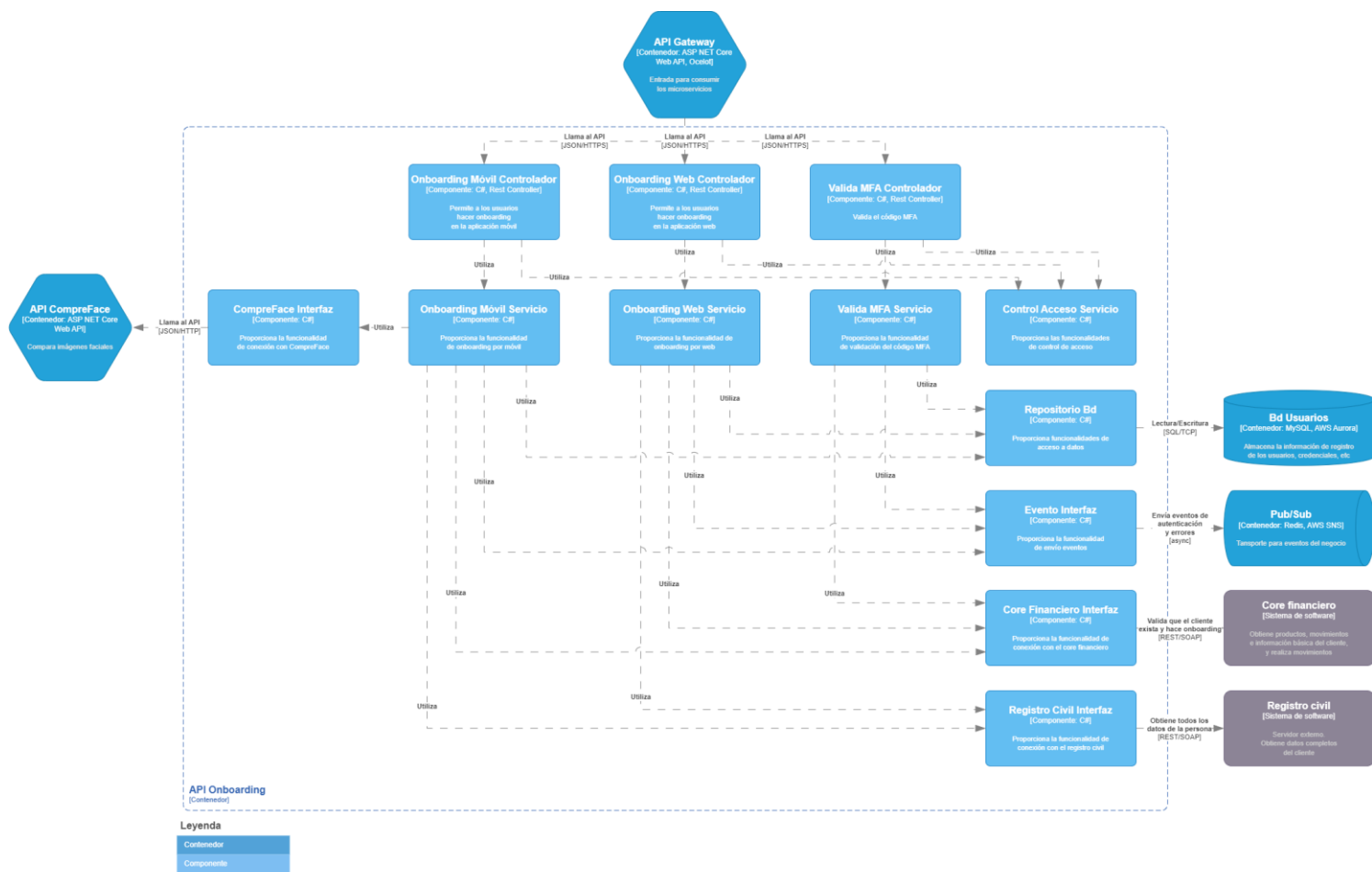
## API Autenticación



[Componente] BP Banca en Línea - API Autenticación

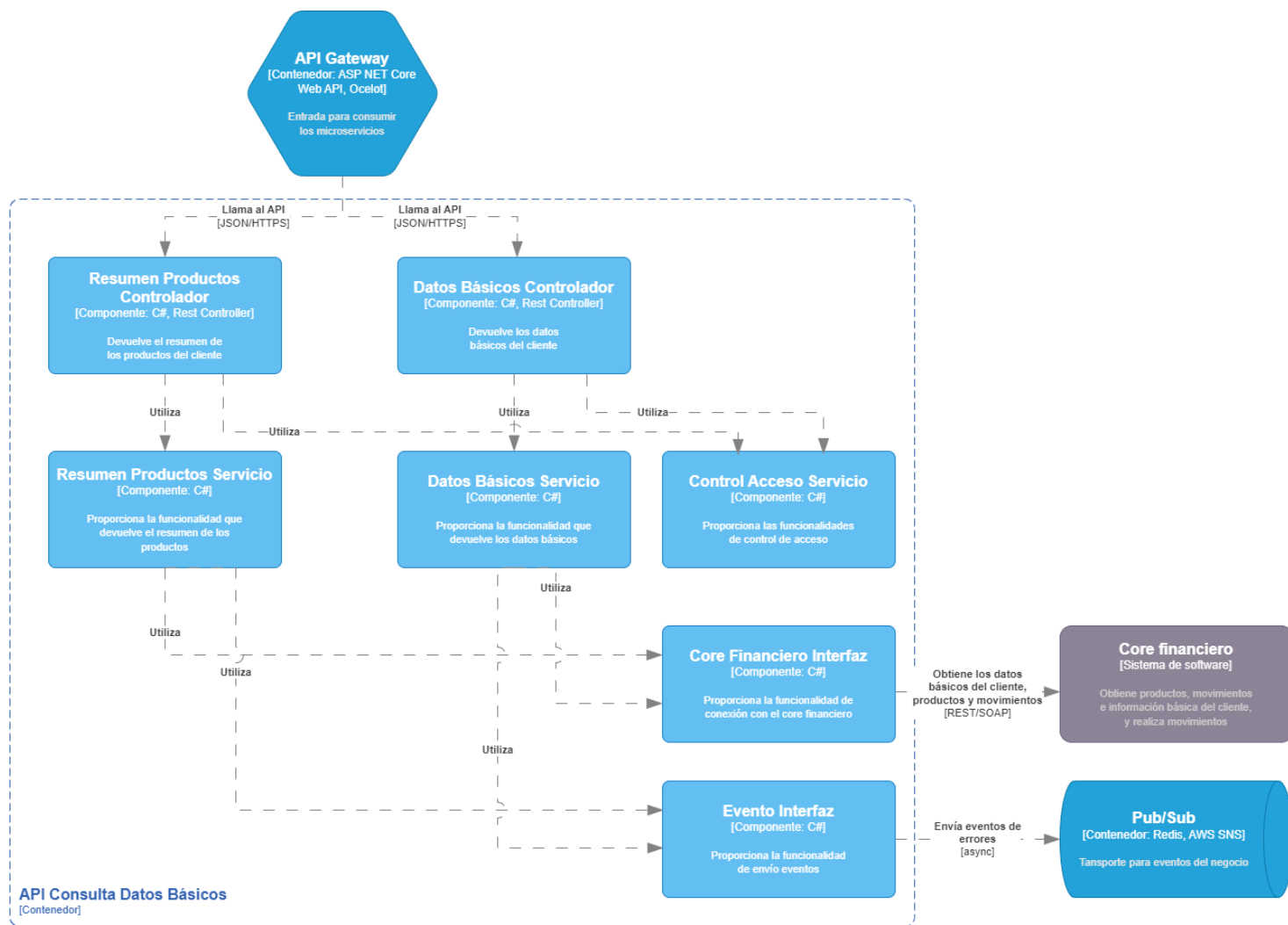


# API Onboarding

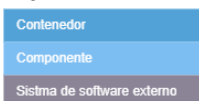


[Componente] BP Banca en Línea - API Onboarding

## API Consulta Datos Básicos

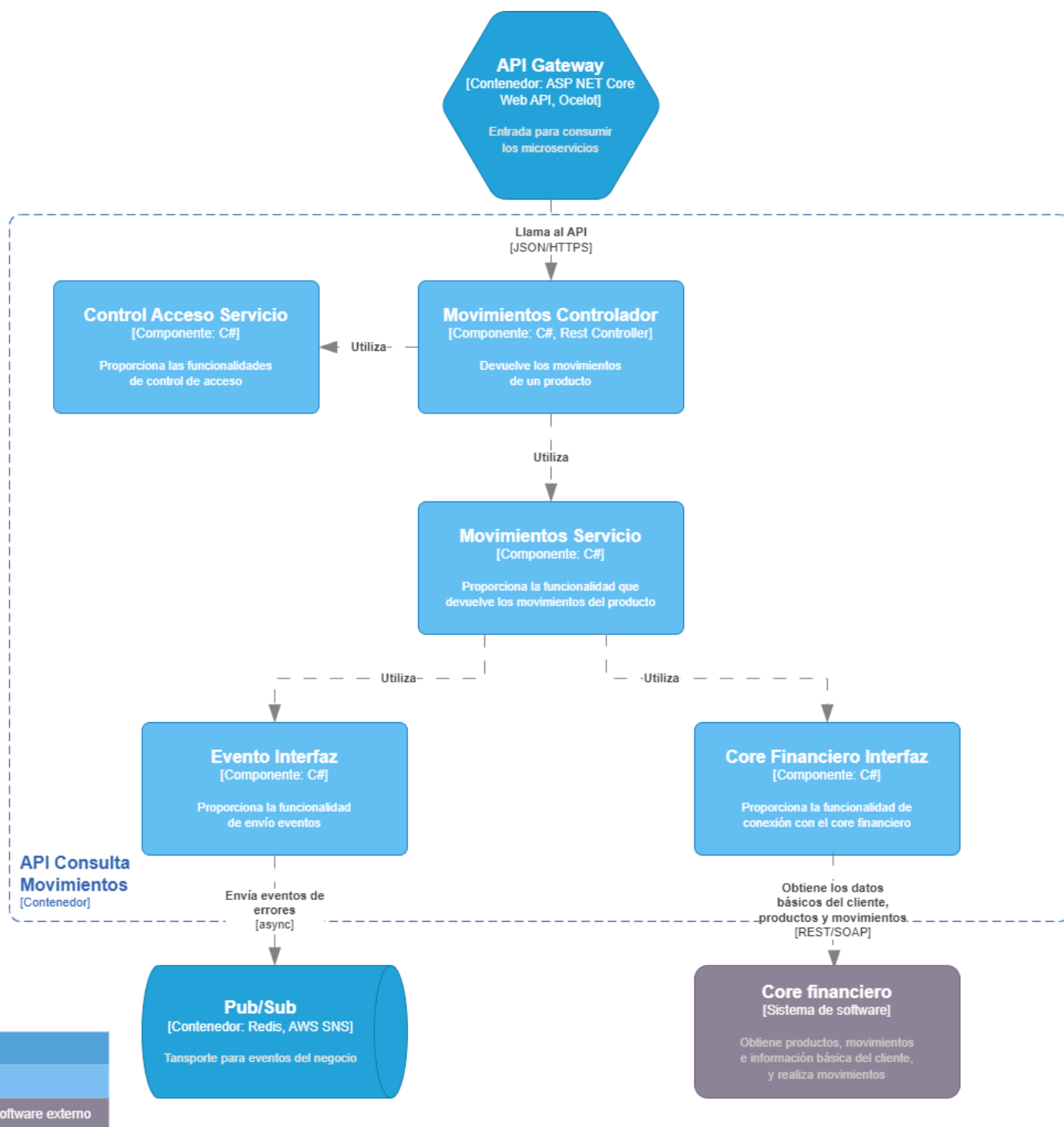


### Leyenda



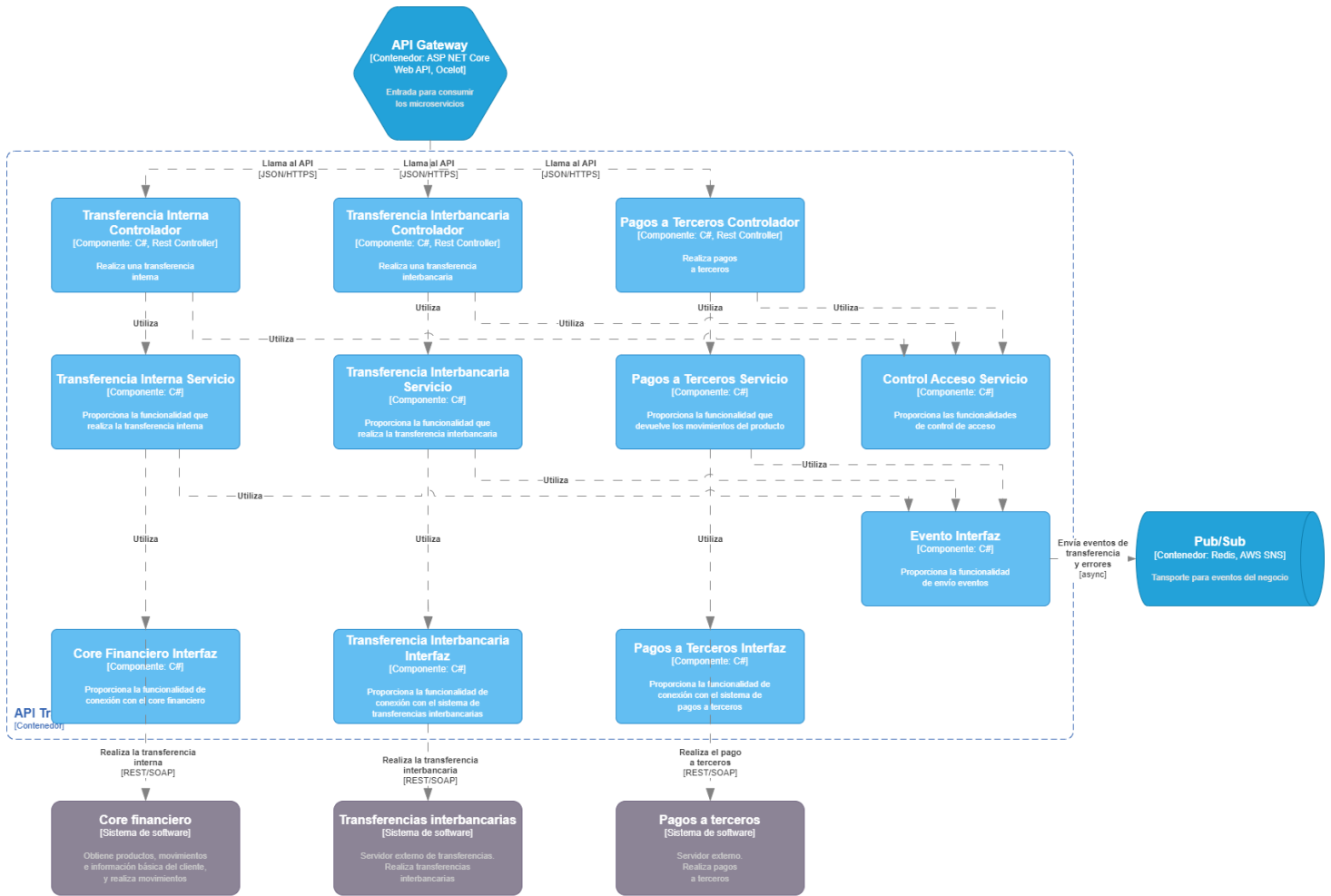
[Componente] BP Banca en Línea - API Consulta Datos Básicos

## API Consulta Movimientos



[Componente] BP Banca en Línea - API Consulta Movimientos

## API Transferencias

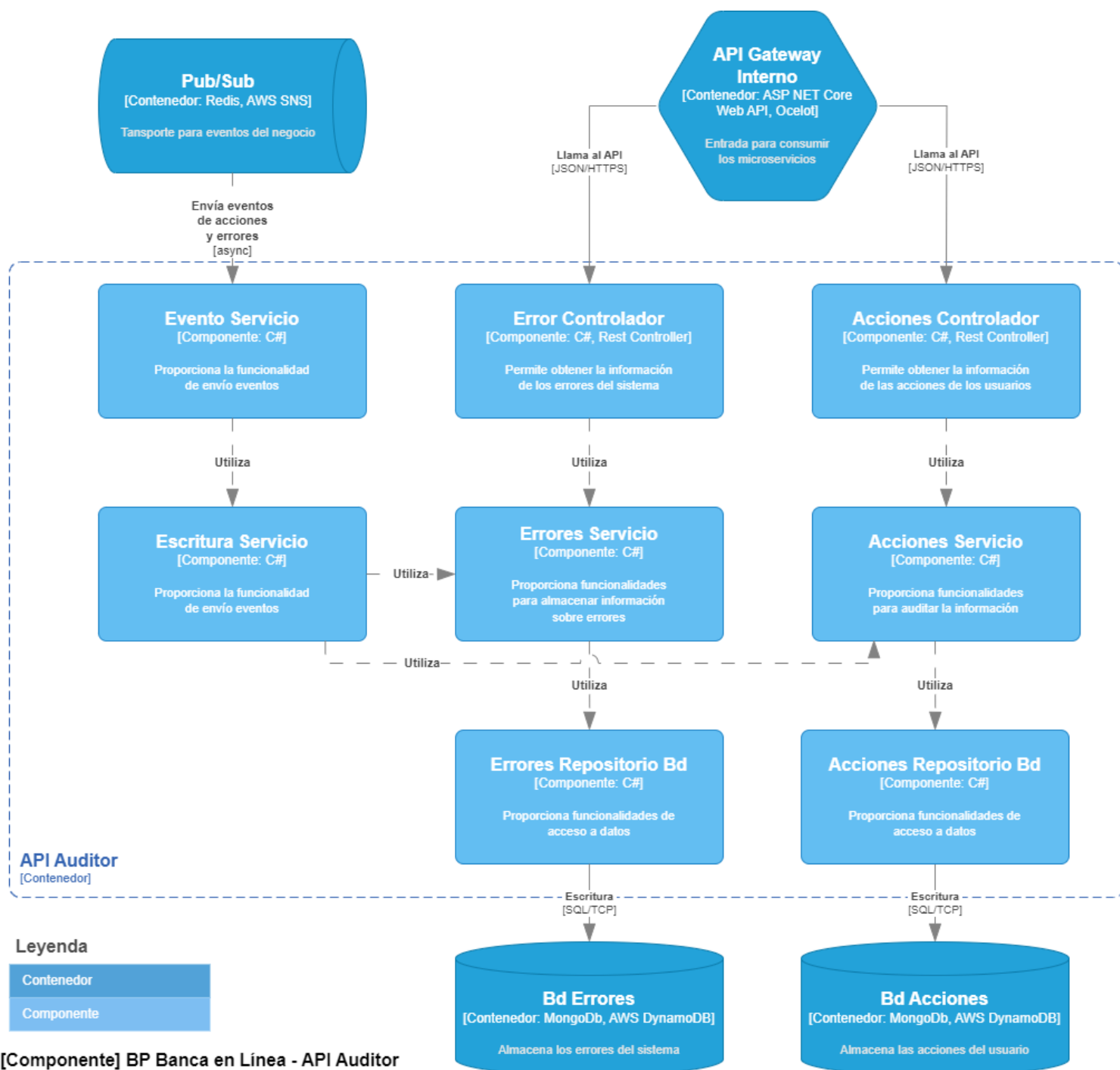


### Legenda

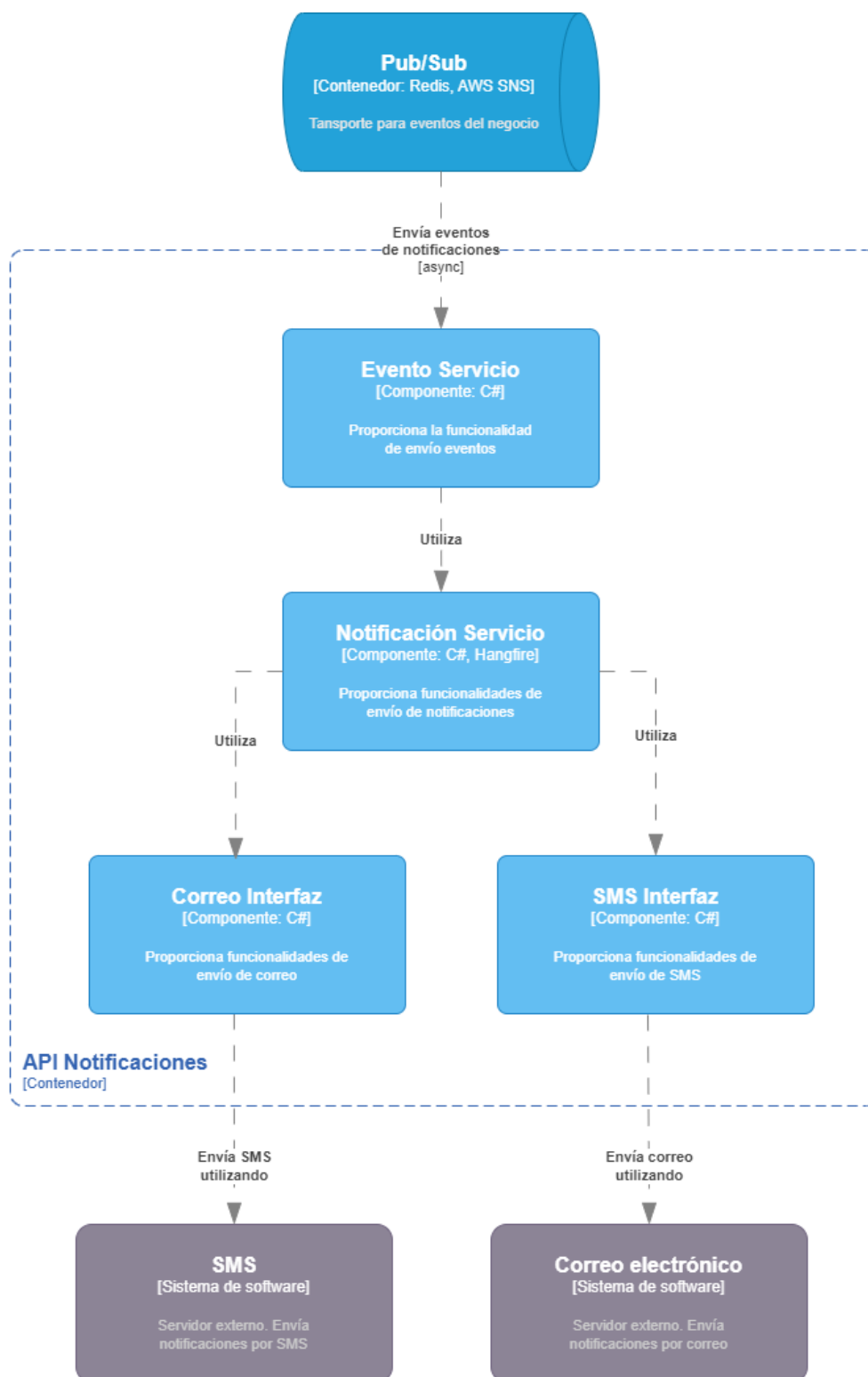
Contenedor
Componente
Sistema de software externo

[Componente] BP Banca en Línea - API Transferencias

## API Auditor



## API Notificaciones

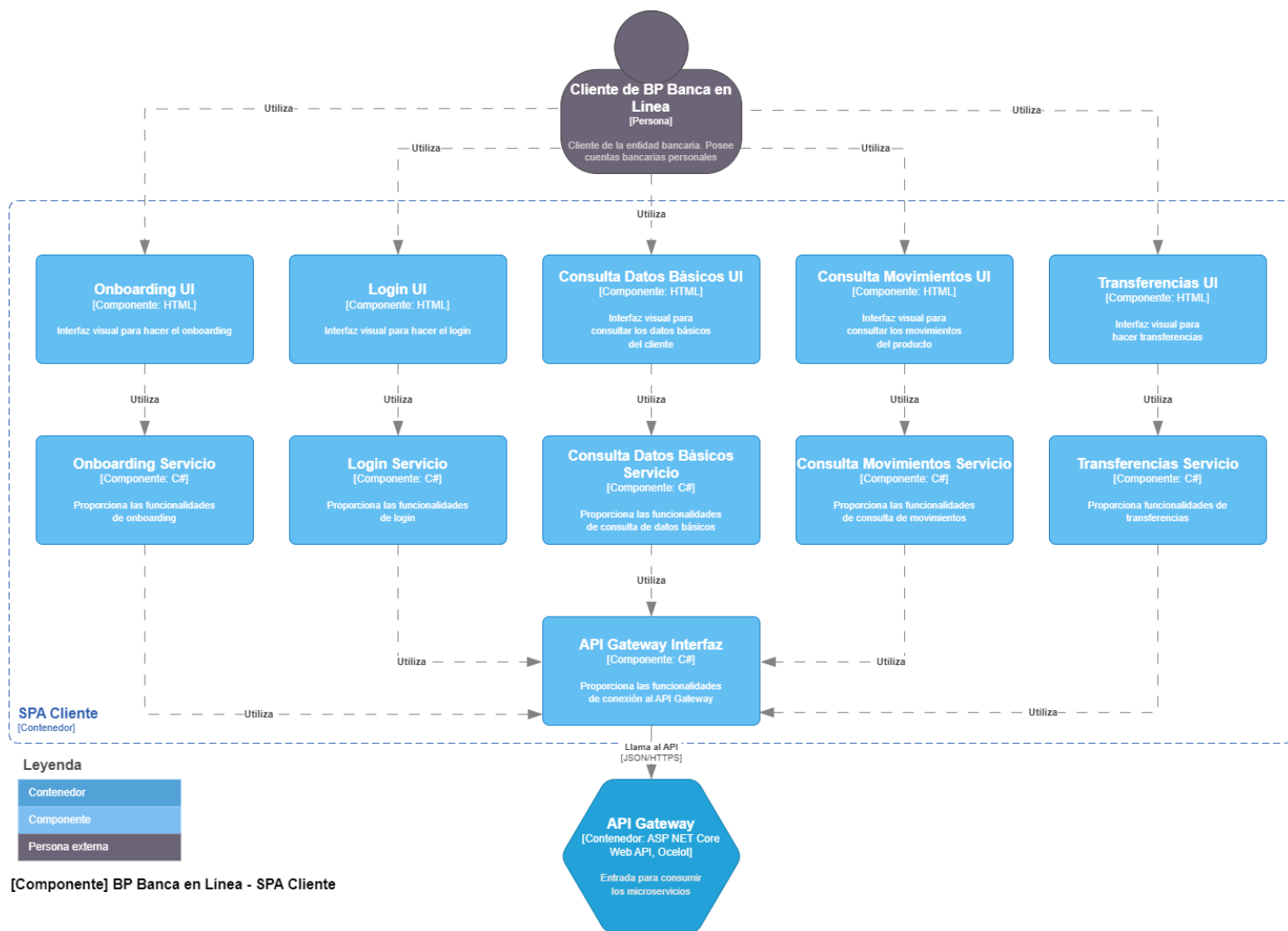


### Leyenda

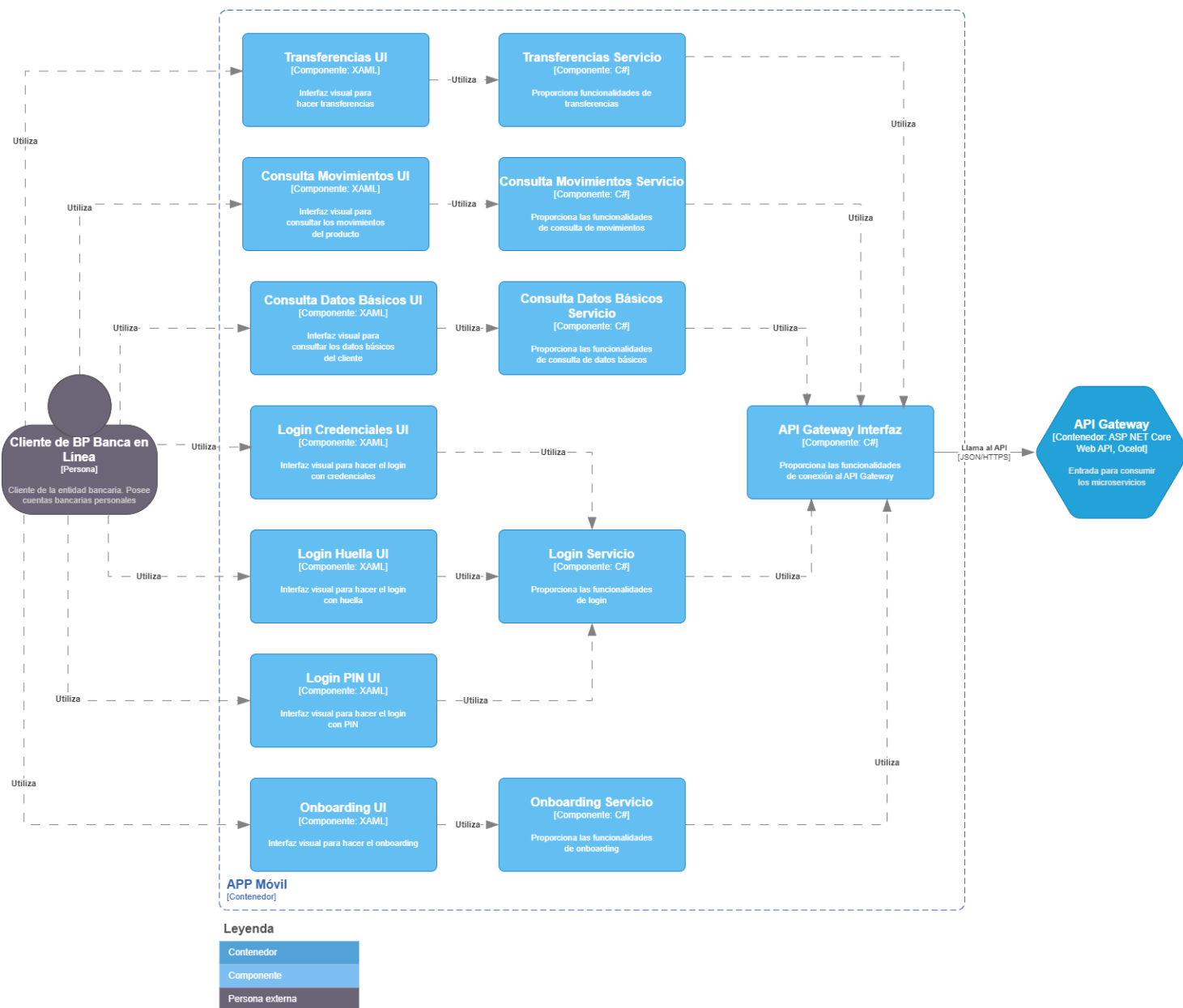
Contenedor
Componente
Sistema de software externo

[Componente] BP Banca en Línea - API Notificaciones

## SPA Cliente



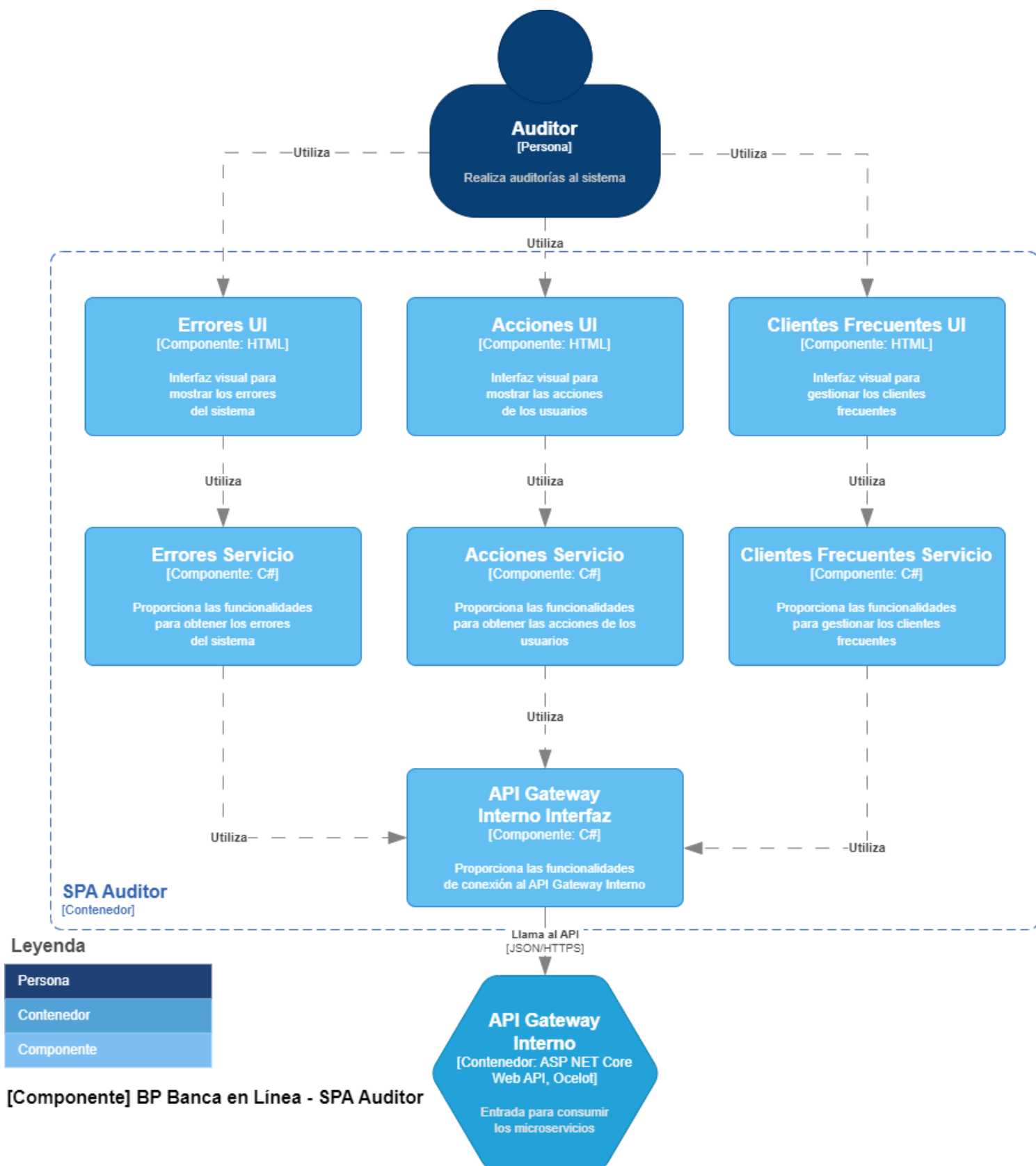
## APP Móvil



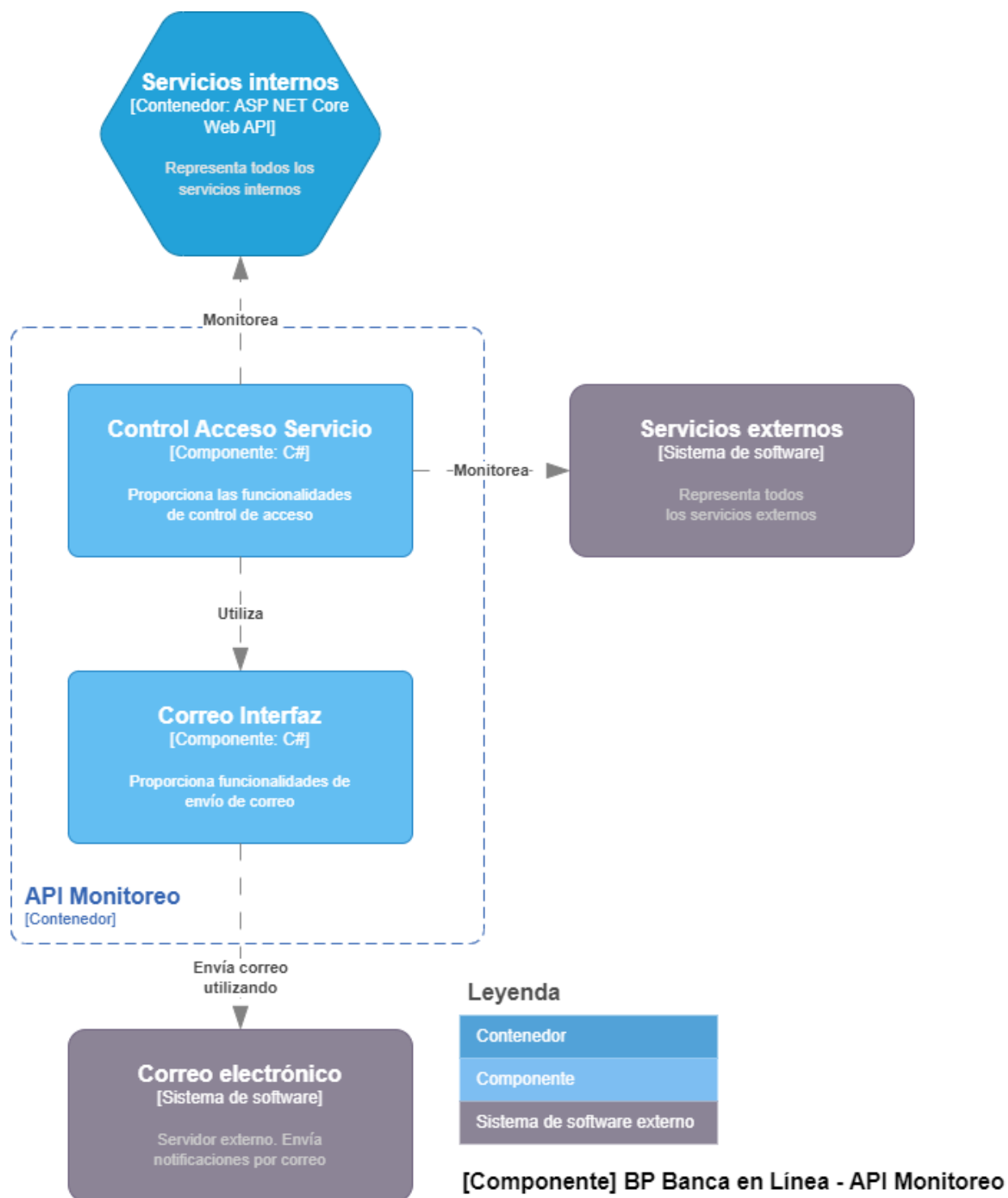
[Componente] BP Banca en Línea - APP Móvil



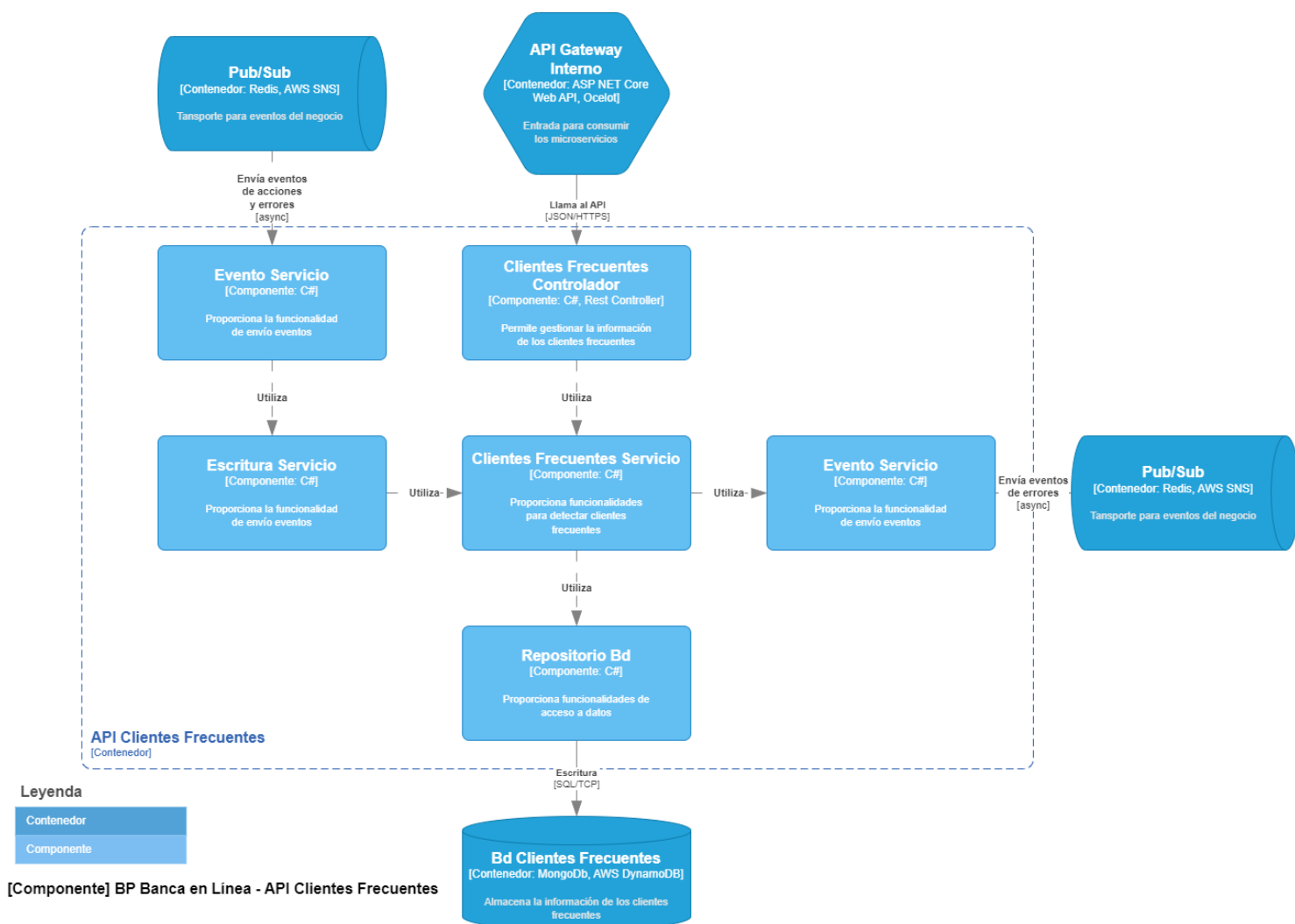
## SPA Auditor



## API Monitoreo



## API Clientes Frecuentes



## Conclusiones

La arquitectura propuesta asegura un sistema de banca en línea robusto, seguro, y preparado para futuras expansiones, alineándose con las mejores prácticas de la industria.