Descripción de la Solución.

Nombre: Msc Rodrigo Vivas
Fecha: : 25/02/2023

Para el análisis de la arquitectura, se considera los siguientes supuestos:

- El cliente posee su arquitectura sobre Net Core desplegados en servidores Linux
- El cliente posee licenciamiento de Oracle Server para base de datos.
- Para los nombres de los proyectos, objetos, y artefactos de este sistema, se utilizará las políticas que el cliente posea. En caso de no disponer de dichas políticas , se usará el estilo de escritura Camel Case.
- Se aplicará SCRUM como metodología de administración del proyecto acorde a los lineamientos de la empresa.

La solución se diseñará acorde a lo siguiente:

- Servidor SPA.- Linux , con IIS Net Core, Angular
- Servidor Backend .- Api Rest Controller, sobre Net Core con microservicios. Se utilizará JWT para asegurar los servicios. Se dispondrá de un servidor Linux con IIS Net Core, el lenguaje a utilizar será C# NET.
- Controlador de versionamiento.- Se utilizara GIT para controlar el versionamiento de código. En caso de disponer Team Foundation Server, se utilizará este sistema. En caso de usar Git, se deberán crear 3 proyectos como mínimo
    - Front End SPA
    - Backend Aplicación
    - Aplicación Bancaria
- Se crearán las siguientes ramas en cada proyecto

    - Develop.- Esta rama se usará para el aquellos artefactos que se encuentren en proceso de desarrollo
    - Testing.- Esta rama se usará para aquellas versiones que se liberan en el entorno de Testing para el equipo de Test de la aplicación. Este equipo de Test debe estar contar con personal del cliente
    - QA. - Esta rama se usará para las versiones que se liberan a QA.
    - Producción. - Rama utilizada para la versión que se libera en producción.

- Base de Datos. - Oracle Data Base, acorde a la infraestructura que posea el Cliente.
- Para el acceso a base de datos de Login se utilizará el patrón de Diseño Repository
- Se usará el Patron Singleton para el Login, con inyección de dependencias.
- Aplicación Móvil. - Se puede utilizar los siguientes Frameworks:
    - Xamarin
    - React Native. Se recomienda usar este Framework para mantener React como Framework de Front End.

Sistema de Autenticación y Autorización.

El cliente posee un Servicio de Autenticación con el estándar Oauth2.0, se debería considerar lo siguiente:

- Utilizar doble factor de autenticación.
- Cifrado de claves al ser almacenadas

Se puede utilizar JWT o Identity de Net Core para asegurar los servicios del Backend

- Debe poseer un servicio de Autenticación, el cual retornará un Token valido. El tiempo de vida del Token no deberá ser mayor a 5 minutos.
- Todos los servicios Web, solicitaran un Token para verificar la autorización de acceso al recurso.

Para el alta de usuarios desde la aplicación Web se utilizará lo siguiente:

- Registro con sistema Biométrico de autenticación. Huellas dactilares o reconocimiento facial.
- En caso de claves, se debe colocar el formato mínimo establecido de seguridad para las claves.
  - Largo mínimo. - 10 caracteres.
  - Complejidad: Debe poseer letras, números, mayúsculas, caracteres especiales.
  - Histórico de claves. - Número de calves históricas que no deben repetirse
- Estos parámetros deben estar alienados a las políticas de seguridad del cliente.

El proceso de alta de clientes será el siguiente:

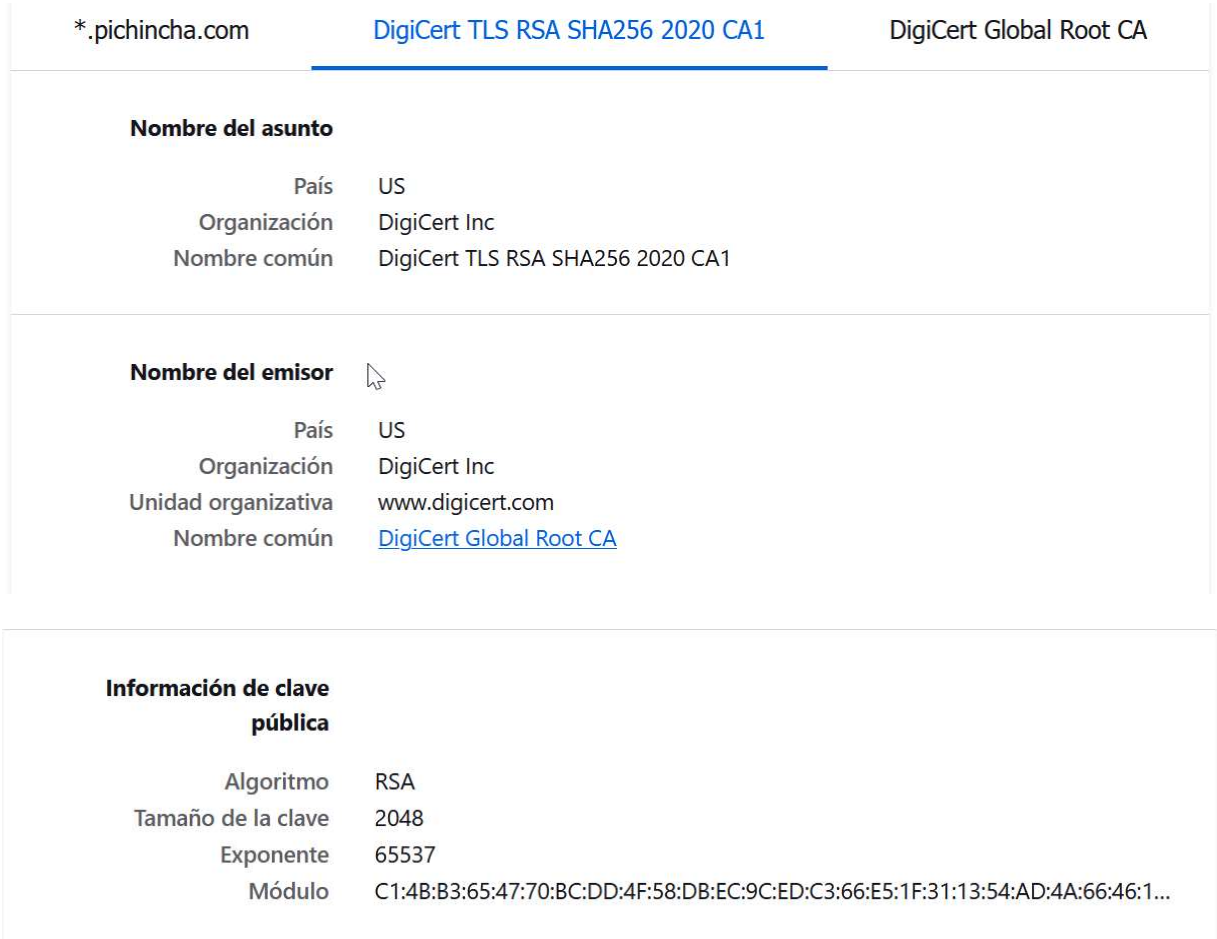
- Creación de usuarios desde la aplicación SPA.
  - Se solicita información de correo electrónico.
  - Se solicita cedula de identidad.
  - El sistema valida que sea un cliente existente en el Banco, envía un código de confirmación al teléfono registrado del cliente
  - El cliente confirma el código recibido.
  - La aplicación SPA solicita información complementaria del cliente
  - Se solicita una clave de seguridad acorde a las políticas establecidas por el cliente
  - El sistema crea el cliente y concede el acceso a la aplicación
  - El sistema registra en la base de datos de Logs toda la información necesaria del proceso (IP origen, fecha, hora, transacción, tipo de navegador, etc)
- Creación de usuarios desde un dispositivo móvil.
  - Se solicita información de correo electrónico.
  - Se solicita cedula de identidad.
  - La aplicación valida que sea un cliente existente en el Banco, envía un código de confirmación al teléfono registrado del cliente
  - Reconocimiento facial)
  - La aplicación confirma identidad de cliente.
  - La aplicación crea al cliente en el sistema
  - El sistema registra el log de auditoria, similar que en el alta de clientes en el SPA

Arquitectura de Seguridad

En caso de usar la infraestructura del cliente, se utilizarán los sistemas de Seguridad que el cliente posea.

Se recomienda lo siguiente:

- Firewall de Aplicaciones Web
- Proxy reverso para la aplicación.
- Certificado de Seguridad para la aplicación, acorde al estándar de seguridad establecido en las Políticas de Seguridad Informática de la entidad Bancaria . Acorde al certificado de seguridad disponible, es posible reutilizar el mismo.



Los elementos normativos para tener en cuenta son:

- Resolución de la Junta Bancaria JB-2012-2148
- Política de Protección de Datos Personales
- Políticas de Seguridad Informática expedidas por el organismo de control (Superintendencia de Bancos)
- Ley de Comercio Electrónico, firmas y mensajes de Datos

Además, se debe considerar lo siguiente:

- Políticas de Seguridad de La información, Políticas informáticas de la entidad Bancaria
- Manual de Protección de Datos Personales de la Entidad Bancaria.
- Normas ISO 27000 de Gestión de la Seguridad de la Información.

En caso de no disponer infraestructura el cliente se podrá evaluar soluciones en Azure, acorde a lo siguiente:

- Para el servidor de Aplicaciones IIS Net Core.- Sistema Operativo Linux Ubuntu / Debian.- Para estimar los recursos es necesario estimar los usuarios concurrentes de la aplicación, esto nos proporcionará los datos para la estimación de memoria, procesador y disco necesarios.
- El monitoreo se lo realizará con Azure Monitor.
- Se debe configurar HA en los servidores de aplicaciones y de base de datos. Para ello se puede habilitar Failover Failback en Azure.
- Para la base de datos se puede aplicar HA a nivel de servidor con Azure o a nivel de Base de datos Oracle.
- Se de implementar un esquema de respaldos de los equipos virtuales acorde a lo siguiente:
    - Respaldos completos diarios, a las 2 am o en el horario en el que por seguridad la aplicación permanezca fuera de servicio.
    - Respaldos incrementales cada hora.
    - El sistema para utilizar es Azure Backup
    - Respaldos de la base de datos completos diariamente e incrementales cada hora.
    - Los respaldos deben enviados a una infraestructura diferente.
- Las políticas de respaldo deben estar alineadas las Políticas establecidas de la entidad bancaria.

Diagrama de Contexto

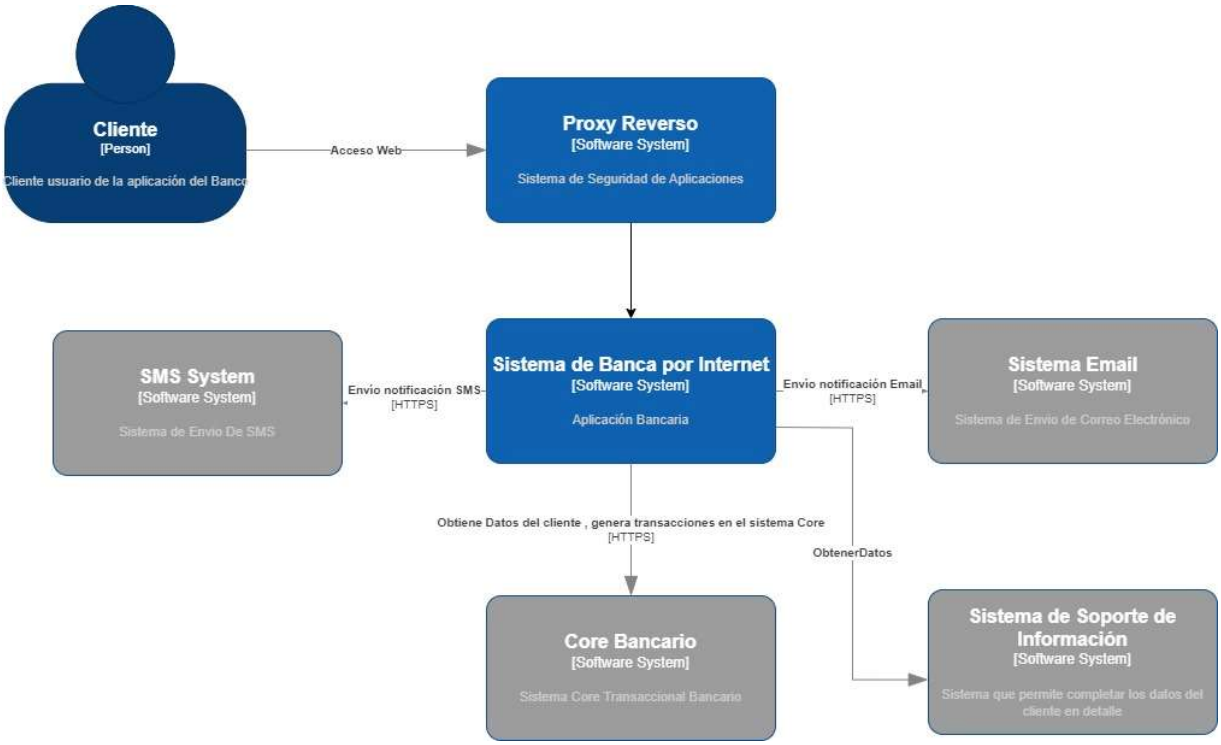


Diagrama de Contenedores

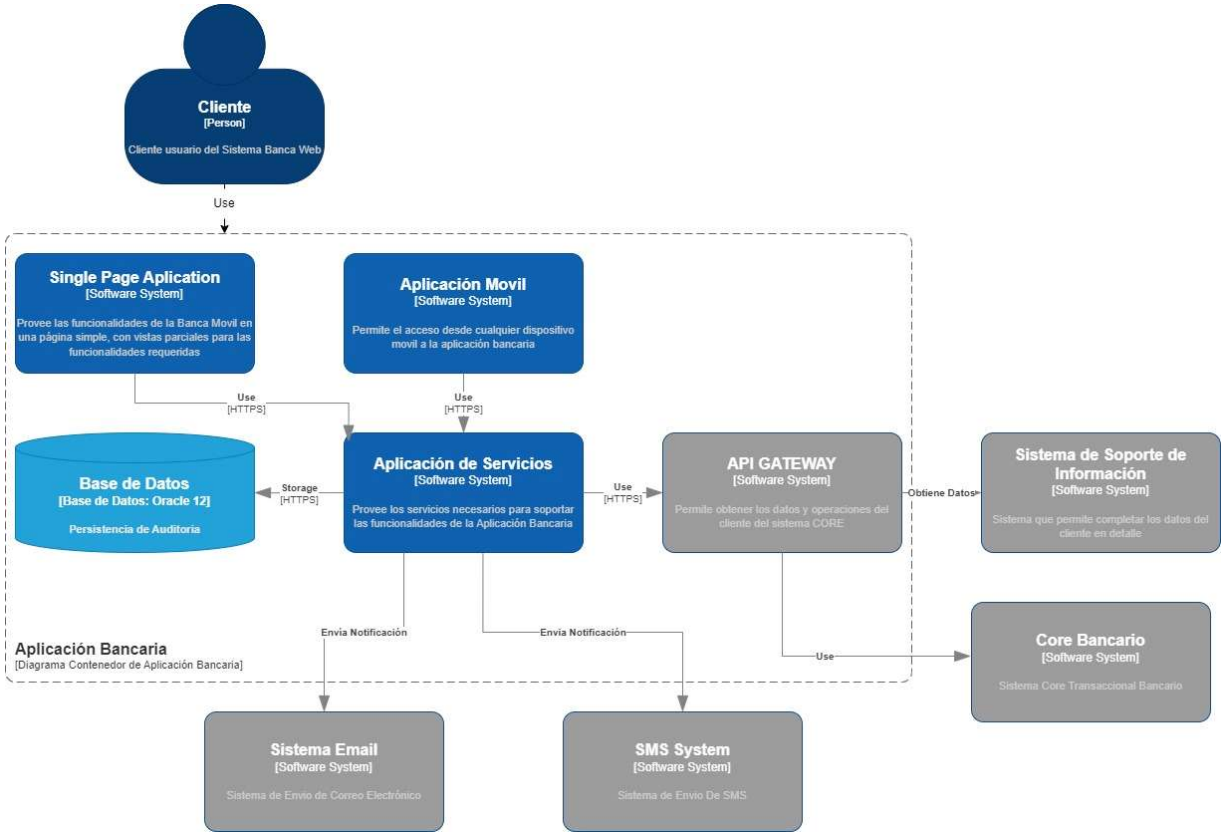
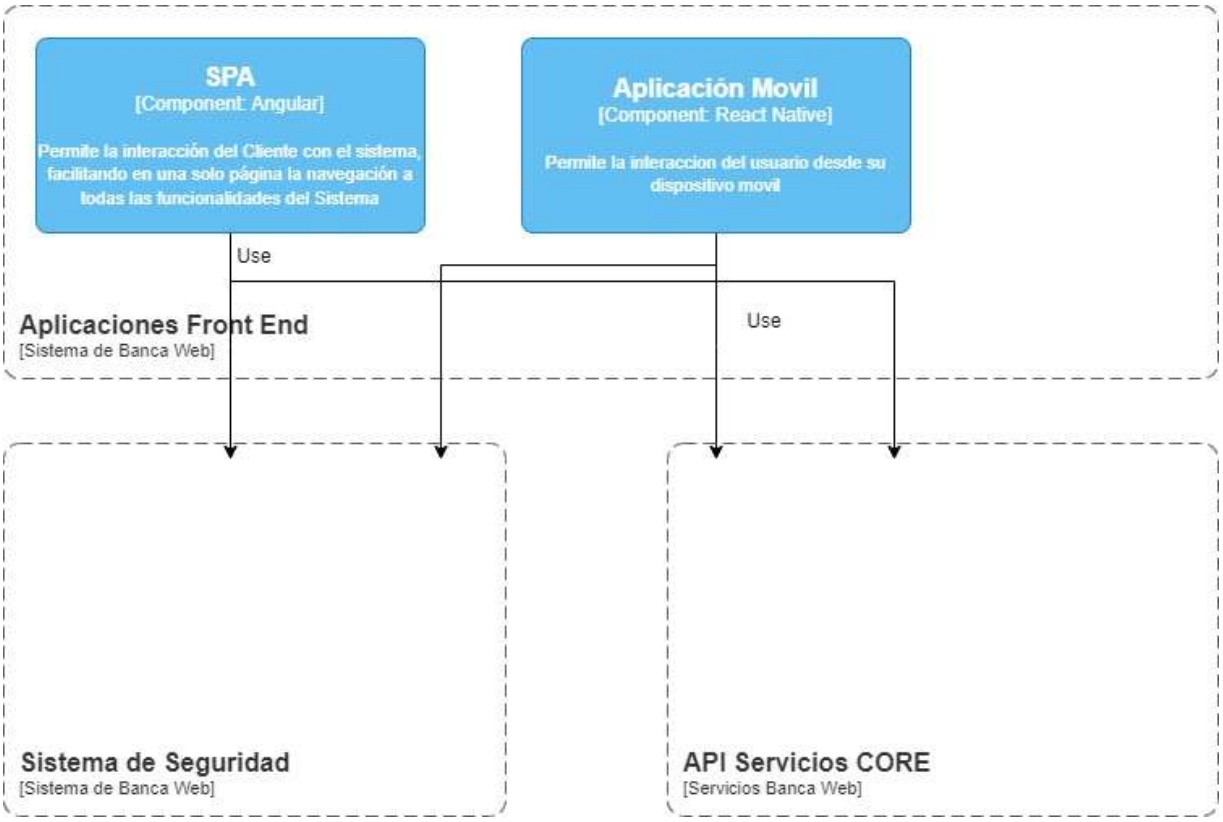
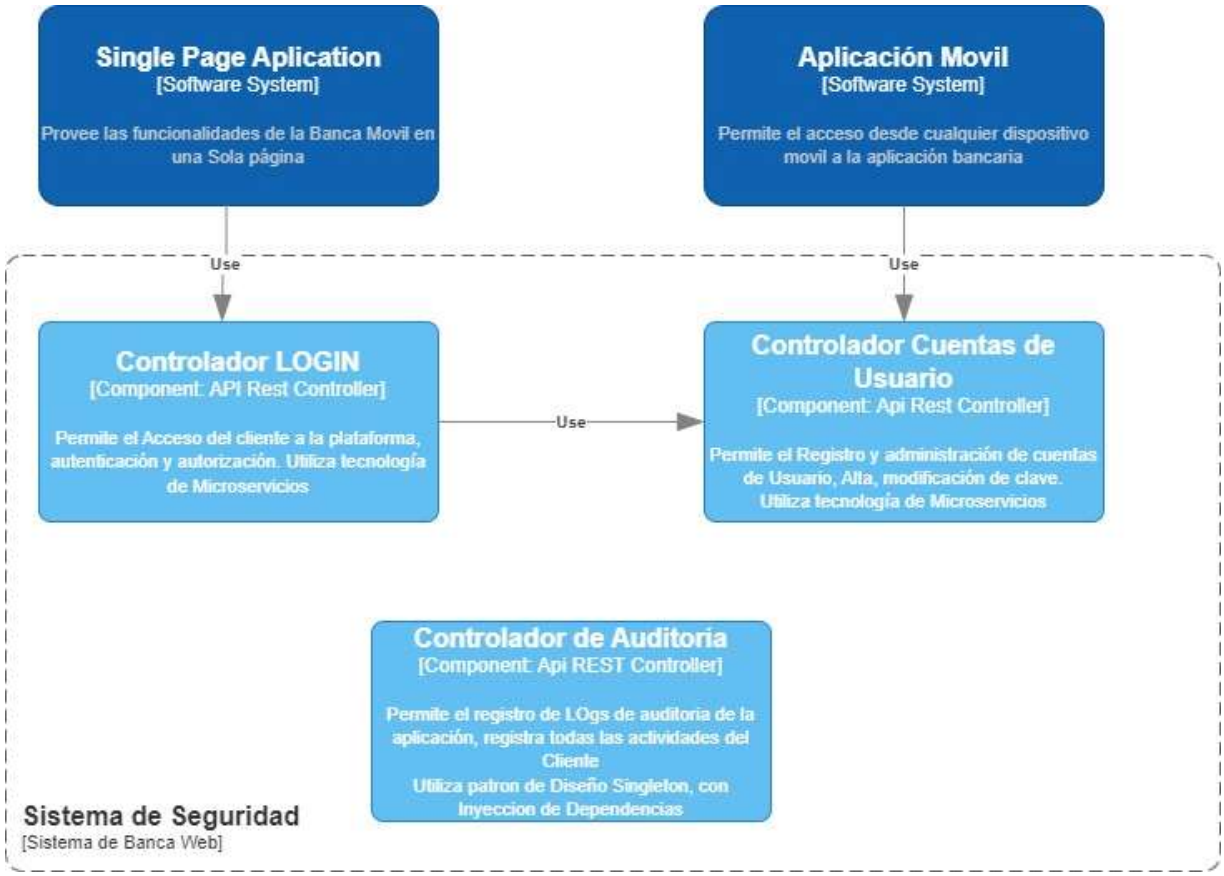
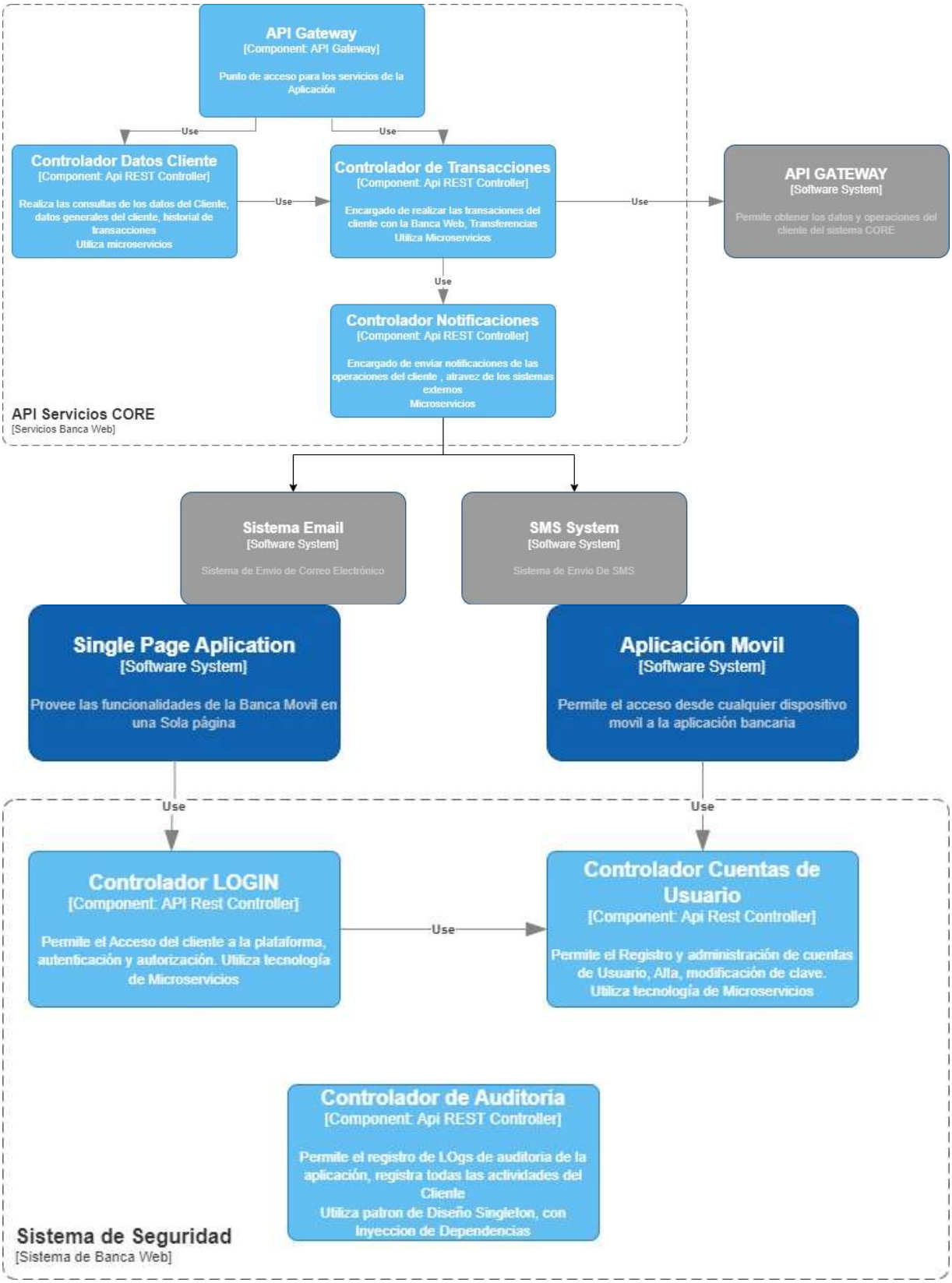


Diagrama de Componentes




Single Page Aplication
[Software System]
Provee las funcionalidades de la Banca Movil en una Sola página
Aplicación Movil
[Software System]
Permite el acceso desde cualquier dispositivo movil a la aplicación bancaria
Use
Use
Controlador LOGIN
[Component: API Rest Controller]
Permite el Acceso del cliente a la plataforma, autenficación y autorización. Utiliza tecnología de Microservicios
Use
Controlador Cuentas de Usuario
[Component: Api Rest Controller]
Permite el Registro y administración de cuentas de Usuario, Alta, modificación de clave. Utiliza tecnología de Microservicios
Controlador de Auditoria
[Component: Api REST Controller]
Permite el registro de LOgs de auditoria de la aplicación, registra todas las actividades del Cliente
Utiliza patron de Diseño Singleton, con Inyeccion de Dependencias
Sistema de Seguridad
[Sistema de Banca Web]


API Gateway
[Component: API Gateway]
Punto de acceso para los servicios de la Aplicación
Use
Use
Controlador Datos Cliente
[Component: Api REST Controller]
Realiza las consultas de los datos del Cliente, datos generales del cliente, historial de transacciones
Utiliza microservicios
Use
Controlador de Transacciones
[Component: Api REST Controller]
Encargado de realizar las transaciones del cliente con la Banca Web, Transferencias
Utiliza Microservicios
Use
API GATEWAY
[Software System]
Permite obtener los datos y operaciones del cliente del sistema CORE
Use
Controlador Notificaciones
[Component: Api REST Controller]
Encargado de enviar notificaciones de las operaciones del cliente , atravez de los sistemas externos
Microservicios
API Servicios CORE
[Servicios Banca Web]
Sistema Email
[Software System]
Sistema de Envio de Correo Electrónico
SMS System
[Software System]
Sistema de Envio De SMS
Single Page Aplication
[Software System]
Provee las funcionalidades de la Banca Movil en una Sola página
Aplicación Movil
[Software System]
Permite el acceso desde cualquier dispositivo movil a la aplicación bancaria
Use
Use
Controlador LOGIN
[Component: API Rest Controller]
Permite el Acceso del cliente a la plataforma, autenticación y autorización. Utiliza tecnología de Microservicios
Use
Controlador Cuentas de Usuario
[Component: Api Rest Controller]
Permite el Registro y administración de cuentas de Usuario, Alta, modificación de clave.
Utiliza tecnología de Microservicios
Controlador de Auditoria
[Component: Api REST Controller]
Permite el registro de LOgs de auditoria de la aplicación, registra todas las actividades del Cliente
Utiliza patron de Diseño Singleton, con Inyeccion de Dependencias
Sistema de Seguridad
[Sistema de Banca Web]

Sistema Onboarding

El sistema Onboarding a usar será orientado a funciones, principalmente las siguientes funciones:

- Creación de usuario en el sistema
- Consulta de Saldos
- Transferencias