14 de febrero de 2025
Ing. Jesús Guerra Cedeño
Jguerra.1991@outlook.com

Reto Técnico Arquitectura de Soluciones

Descripción/objetivo del producto

Reto técnico para aplicar al cargo de Arquitecto de Soluciones para Devsu, para proyectos de core bancario y financiero.

Antecedentes

Para la elaboración del reto técnico se han establecido ciertos requerimientos y lineamientos para poner a prueba al aspirante del cargo de Arquitecto de Soluciones sus destrezas y habilidades. La descripción técnica del requerimiento se encuentra descrita en la plataforma de evaluaciones de Devsu.

Descripción técnica de la propuesta

En este apartado se describen los modelos propuestos para la Arquitectura planteada

1. Modelo de Contexto

En este modelo identificamos los actores principales y sus interacciones con el sistema:

- **Usuarios:** Clientes del banco que acceden al sistema a través de la aplicación móvil o la SPA (Single Page Application).
- **Sistema de Banca por Internet:** El sistema que permite a los usuarios realizar consultas y transacciones.
- **Core Bancario:** Sistema que contiene la información básica de los clientes y sus productos.
- **Sistema Independiente:** Sistema que complementa la información del cliente cuando es necesario.
- **Sistemas de Notificaciones:** Plataformas externas o internas que notifican a los usuarios sobre sus movimientos.
- **API Gateway:** Capa de integración que gestiona el tráfico entre los front-ends y los servicios de backend.

2. Modelo de Aplicación / Contenedor

2.1. *Aplicación Móvil*

- *Frameworks recomendados:*
  a. **Flutter:** Versión 3.x. Destaca por su alta performance, personalización, y soporte para múltiples plataformas desde un solo código base.
  b. **React Native:** Versión 0.71.x. Ideal por su integración con el ecosistema JavaScript y su comunidad robusta.

- *Autenticación biométrica:*
  c. **Plugins y SDKs**:
     - **Local Authentication (Flutter):** Para huellas digitales y reconocimiento facial.
     - **React Native Biometrics:** Compatible con huellas digitales, reconocimiento facial y otros mecanismos de autenticación biométrica.

2.2. *SPA (Single Page Application)*

- *Frameworks recomendados:*
  d. **Angular:** Versión 15.x. Proporciona una arquitectura bien definida con herramientas integradas para el manejo de grandes aplicaciones.
  e. **React:** Versión 18.x. Ofrece flexibilidad y un ecosistema robusto, ideal para aplicaciones dinámicas y escalables.

2.3. *Servicio de Autenticación (OAuth 2.0)*

- *Flujo recomendado:*
  f. **Authorization Code Flow**: Es el más seguro y adecuado para aplicaciones web y móviles, ya que no expone tokens de acceso directamente al cliente.

- *Herramientas recomendadas:*
  g. **Keycloak**: Versión 20.x. Proveedor de identidad y accesos que soporta OAuth 2.0, OpenID Connect y SAML.
  h. **Okta:** Proveedor comercial de identidad que ofrece soporte robusto para OAuth 2.0 y flujos avanzados de autenticación.

2.4. *Servicio de Onboarding con Reconocimiento Facial*

- *Herramientas recomendadas:*
  i. **Amazon Rekognition:** Servicio gestionado de AWS para análisis de imágenes y vídeos que incluye capacidades de reconocimiento facial.
  j. **Microsoft Azure Face API:** Servicio de Azure para reconocimiento facial, altamente preciso y escalable.

- *Implementación del Reconocimiento Facial:*

Durante el onboarding, el usuario tomará una foto o escaneará su rostro, que será comparado con documentos oficiales o fotos previas almacenadas en el sistema. Este proceso asegura que solo el titular legítimo del dispositivo y la cuenta pueda completar el registro.

2.5. *Base de Datos Principal (Core Bancario)*

- *Herramientas recomendadas:*
  k. **Oracle Database:** Versión 19c. Altamente confiable para manejar transacciones bancarias con soporte avanzado para replicación y alta disponibilidad.
  l. **PostgreSQL:** Versión 14.x. Una opción de código abierto robusta y escalable, con soporte para transacciones ACID.

2.6. *Base de Datos Independiente*

- *Herramientas recomendadas:*
  m. **MySQL:** Versión 8.x. Popular por su simplicidad y eficacia en la gestión de datos complementarios.
  n. **MongoDB:** Versión 6.x. Adecuada para almacenamiento flexible de datos no estructurados o semi-estructurados.

2.7. *Base de Datos de Auditoría*

- *Herramientas recomendadas:*
  o. **MongoDB:** Versión 6.x. Adecuada para almacenar logs de auditoría de forma eficiente.

p. **Amazon DynamoDB:** Proporciona una base de datos NoSQL completamente gestionada y escalable, ideal para auditoría y almacenamiento de eventos.

2.8. API Gateway

- *Herramientas recomendadas:*
  q. **AWS API Gateway:** Servicio gestionado de AWS que facilita la creación y gestión de APIs a escala.
  r. **Kong Gateway:** Versión 3.x. Un gateway de código abierto que ofrece alta flexibilidad, seguridad, y performance para gestionar microservicios.

3. Modelo de Componentes

- **Frontend (SPA y Aplicación Móvil):** Se conectan con el API Gateway para realizar consultas y transacciones.
- **API Gateway:** Dirige el tráfico a los microservicios correspondientes:
  - **Microservicio de Consulta de Datos Básicos:** Interactúa con el Core Bancario.
  - **Microservicio de Consulta de Movimientos:** Recupera información de la base de datos de movimientos.
  - **Microservicio de Transferencias:** Maneja las transferencias y pagos entre cuentas propias e interbancarias.

3.1. Servicios de Notificaciones

s. **Herramientas recomendadas:**
- **Twilio:** Versión 8.x. Utilizado para el envío de SMS y llamadas de voz, ideal para notificaciones críticas.
- **Amazon SNS (Simple Notification Service):** Servicio gestionado de AWS que permite enviar notificaciones push, mensajes SMS y correo electrónico.

Consideraciones Normativas

- **Ley de Protección de Datos Personales:** Cumplimiento con las normativas locales y LOPDP para el manejo de datos personales. o
- **PCI DSS:** Certificación requerida para el manejo seguro de transacciones con tarjetas de crédito.
- **Normas de Seguridad Informática:** Implementación de cifrado de datos, firewalls, y monitoreo constante para proteger la integridad del sistema.

Garantías de la Arquitectura

- **Alta Disponibilidad (HA):** Se implementará a través de múltiples zonas de disponibilidad en la nube (AWS/Azure), con balanceo de carga y replicación de bases de datos.
- **Tolerancia a Fallos y DR:** Uso de arquitecturas multi-región y respaldo automatizado en la nube para recuperación ante desastres.
- **Seguridad y Monitoreo:** Se utilizarán servicios de monitoreo como AWS CloudWatch o Azure Monitor, junto con herramientas de seguridad como AWS WAF o Azure Security Center.
- **Auto-healing:** Se configurarán autoescalado y reinicio automático de instancias en caso de fallos.

Infraestructura en la Nube

- **AWS o Azure:** Se recomendará utilizar AWS o Azure por su capacidad de garantizar baja latencia, alta disponibilidad y herramientas avanzadas de seguridad.

Diagramación

1. Diagrama de Contexto (Nivel 1)

El diagrama de contexto proporciona una vista de alto nivel de los actores principales y sus interacciones con el sistema. (pag.10) (anexo código UML: diagrama_contexto.puml).

2. Diagrama de interrelación de componentes (API Gateway, Microservicios, Base de Datos y Autenticación Biométrica)

El diagrama de contexto proporciona una vista de alto nivel de los actores principales y sus interacciones con el sistema. (pag.11) (anexo código UML: diagrama_interrelacion_componentes.puml).

Fuente UML:

1. Diagrama de Contexto (Nivel 1)

```
@startuml
!define RECTANGLE class
!define PACKAGE package
!define COMPONENT component
!define CLOUD cloud
!define DATABASE database
!define ACTOR actor

title Diagrama de Contexto - Sistema de Banca por Internet
actor "Usuario" as User <<Client>>


rectangle "Sistema de Banca por Internet" as BankingSystem {

    package "Frontend" {
        component "Aplicación Móvil" as MobileApp <<Mobile App>>
        component "SPA" as SPA <<Web App>>
    }

    package "Backend" {
        component "API Gateway" as APIGateway <<Integration Layer>>
        database "Core Bancario" as CoreBanking <<Backend System>>
        database "Sistema Independiente" as IndependentSystem <<Backend System>>
        component "Servicio de Autenticación (OAuth 2.0)" as AuthService <<Auth Service>>
        cloud "Servicios de Notificaciones" as NotificationService <<External Service>>
        component "Microservicio de Consulta de Datos Básicos" as BasicDataService <<Microservice>>
        component "Microservicio de Consulta de Movimientos" as MovementsService <<Microservice>>
        component "Microservicio de Transferencias" as TransfersService <<Microservice>>
        component "Servicio de Reconocimiento Facial" as FacialRecognition <<Facial Recognition>>
        database "Base de Datos de Auditoría" as AuditDB <<Database>>
    }
}

User --> MobileApp : Usa User --> SPA : Usa
MobileApp --> APIGateway : Conecta a
SPA --> APIGateway : Conecta a

APIGateway --> BasicDataService : Solicita Datos
```

```
APIGateway --> MovementsService : Solicita Movimientos
APIGateway --> TransfersService : Realiza Transferencias
APIGateway --> AuthService : Autenticación OAuth 2.0
APIGateway --> NotificationService : Envía Notificaciones
APIGateway --> FacialRecognition : Onboarding con Reconocimiento Facial

BasicDataService --> CoreBanking : Consulta Datos
MovementsService --> CoreBanking : Consulta Movimientos
TransfersService --> CoreBanking : Ejecuta Transferencia
CoreBanking --> IndependentSystem : Consulta Datos Adicionales
CoreBanking --> AuditDB : Registra Eventos
TransfersService --> AuditDB : Registra Transferencia

@enduml
```

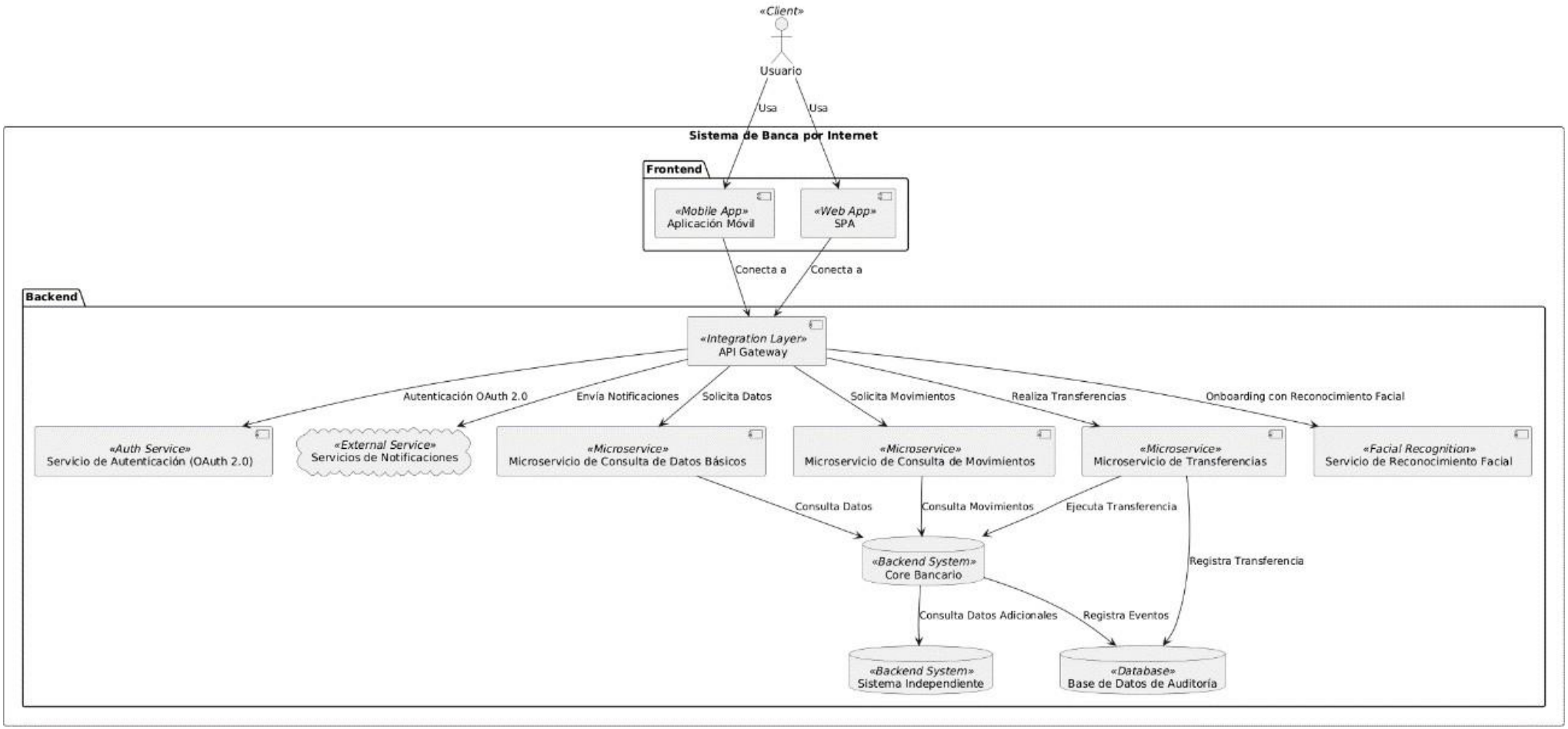
2. Diagrama de interrelación de componentes (API Gateway, Microservicios, Base de Datos y Autenticación Biométrica)

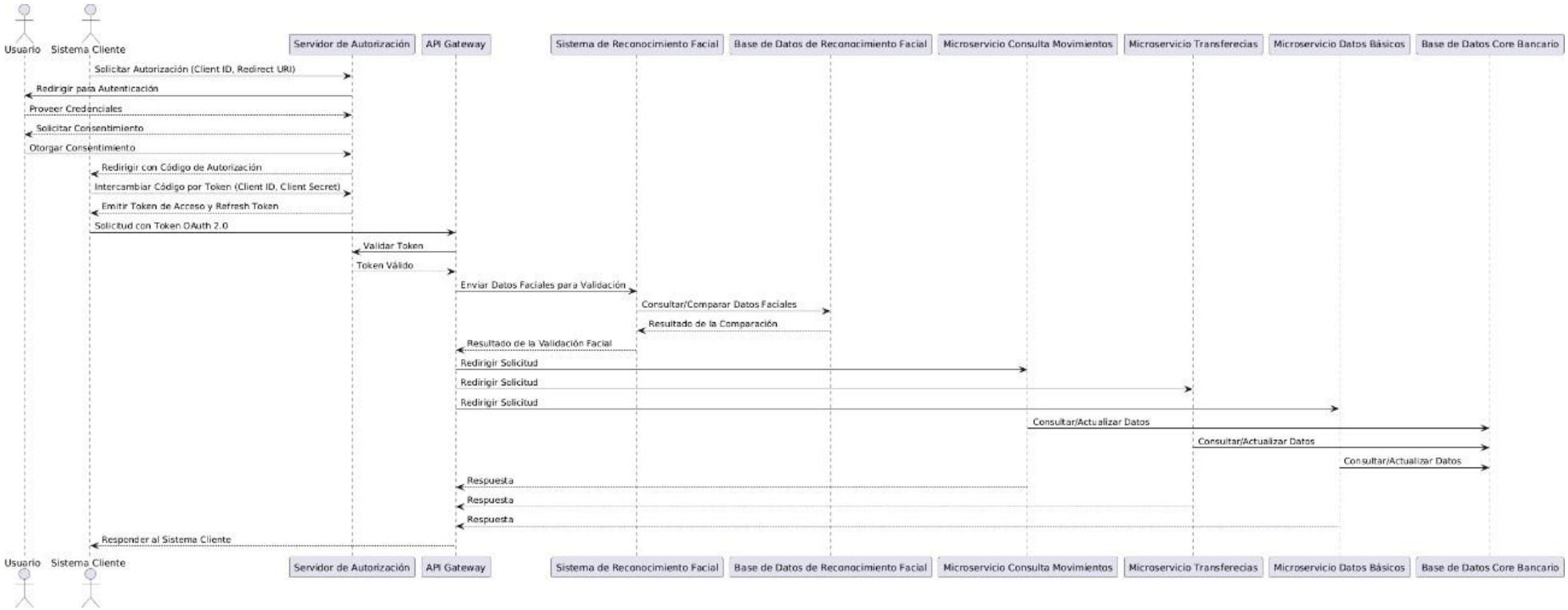
```
@startuml



actor "Usuario" as User

actor "Sistema Cliente" as Client



Client -> "Servidor de Autorización" : Solicitar Autorización (Client ID, Redirect URI)

"Servidor de Autorización" -> User : Redirigir para Autenticación

User --> "Servidor de Autorización" : Proveer Credenciales

"Servidor de Autorización" --> User : Solicitar Consentimiento

User -> "Servidor de Autorización" : Otorgar Consentimiento

"Servidor de Autorización" --> Client : Redirigir con Código de Autorización
```

```
Client -> "Servidor de Autorización" : Intercambiar Código por Token (Client ID, Client Secret)

"Servidor de Autorización" --> Client : Emitir Token de Acceso y Refresh Token



Client -> "API Gateway" : Solicitud con Token OAuth 2.0

"API Gateway" -> "Servidor de Autorización" : Validar Token

"Servidor de Autorización" --> "API Gateway" : Token Válido



"API Gateway" -> "Sistema de Reconocimiento Facial" : Enviar Datos Faciales para Validación

"Sistema de Reconocimiento Facial" -> "Base de Datos de Reconocimiento Facial" : Consultar/Comparar
Datos Faciales

"Base de Datos de Reconocimiento Facial" --> "Sistema de Reconocimiento Facial" : Resultado de la
Comparación

"Sistema de Reconocimiento Facial" --> "API Gateway" : Resultado de la Validación Facial



"API Gateway" -> "Microservicio Consulta Movimientos" : Redirigir Solicitud "API Gateway" ->
"Microservicio Transferecias" : Redirigir Solicitud

"API Gateway" -> "Microservicio Datos Básicos" : Redirigir Solicitud



"Microservicio Consulta Movimientos" -> "Base de Datos Core Bancario" : Consultar/Actualizar Datos

"Microservicio Transferecias" -> "Base de Datos Core Bancario" : Consultar/Actualizar Datos

"Microservicio Datos Básicos" -> "Base de Datos Core Bancario" : Consultar/Actualizar Datos



"Microservicio Consulta Movimientos" --> "API Gateway" : Respuesta
```

```
"Microservicio Transferecias" --> "API Gateway" : Respuesta

"Microservicio Datos Básicos" --> "API Gateway" : Respuesta



"API Gateway" --> Client : Responder al Sistema Cliente



@enduml
```

Diagrama de Contexto - Sistema de Banca por Internet




Usuario
Sistema Cliente
Servidor de Autorización
API Gateway
Sistema de Reconocimiento Facial
Base de Datos de Reconocimiento Facial
Microservicio Consulta Movimientos
Microservicio Transferecias
Microservicio Datos Básicos
Base de Datos Core Bancario
Solicitar Autorización (Client ID, Redirect URI)
Redirigir para Autenticación
Proveer Credenciales
Solicitar Consentimiento
Otorgar Consentimiento
Redirigir con Código de Autorización
Intercambiar Código por Token (Client ID, Client Secret)
Emitir Token de Acceso y Refresh Token
Solicitud con Token OAuth 2.0
Validar Token
Token Válido
Enviar Datos Faciales para Validación
Consultar/Comparar Datos Faciales
Resultado de la Comparación
Resultado de la Validación Facial
Redirigir Solicitud
Redirigir Solicitud
Redirigir Solicitud
Consultar/Actualizar Datos
Consultar/Actualizar Datos
Consultar/Actualizar Datos
Respuesta
Respuesta
Respuesta
Responder al Sistema Cliente