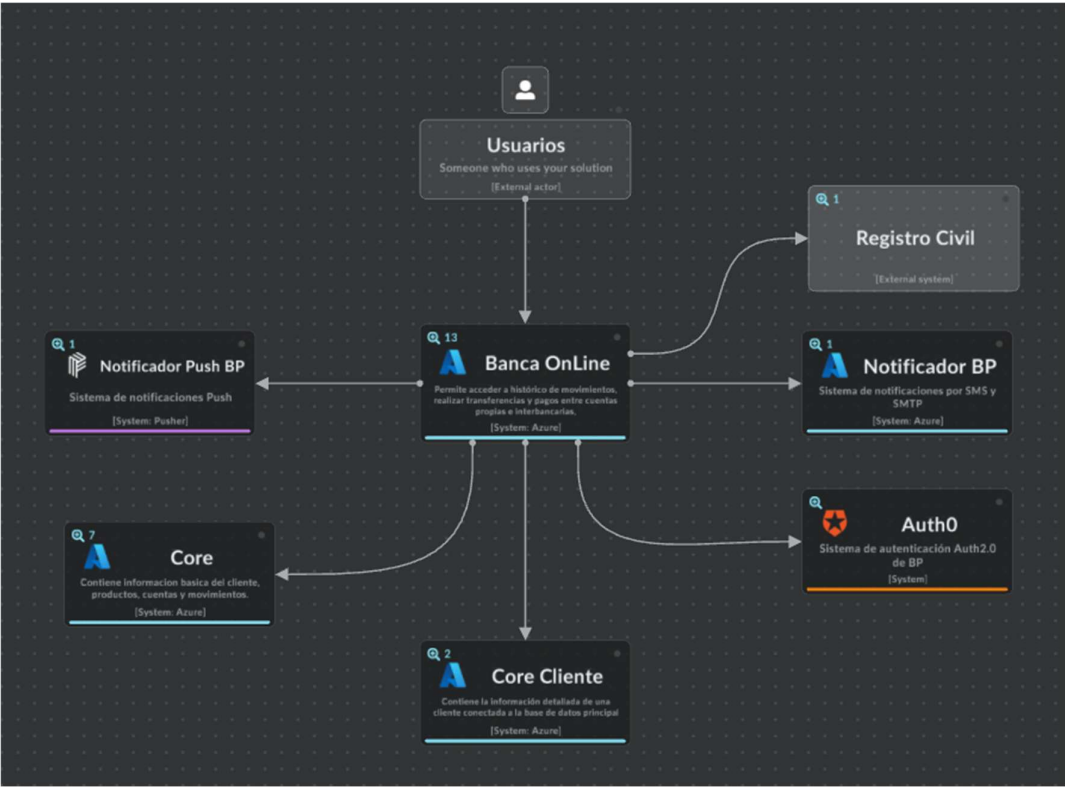


Usted ha sido contratado por una entidad llamada BP como arquitecto de soluciones para diseñar un sistema de banca por internet, en este sistema los usuarios podrán acceder al histórico de sus movimientos, realizar transferencias y pagos entre cuentas propias e interbancarias.

Toda la información referente al cliente se tomará de 2 sistemas, una plataforma Core que contiene información básica de cliente, productos, cuentas, movimientos, y un sistema independiente que complementa la información del cliente cuando los datos se requieren en detalle este sistema core está conectado a una base de datos principal.



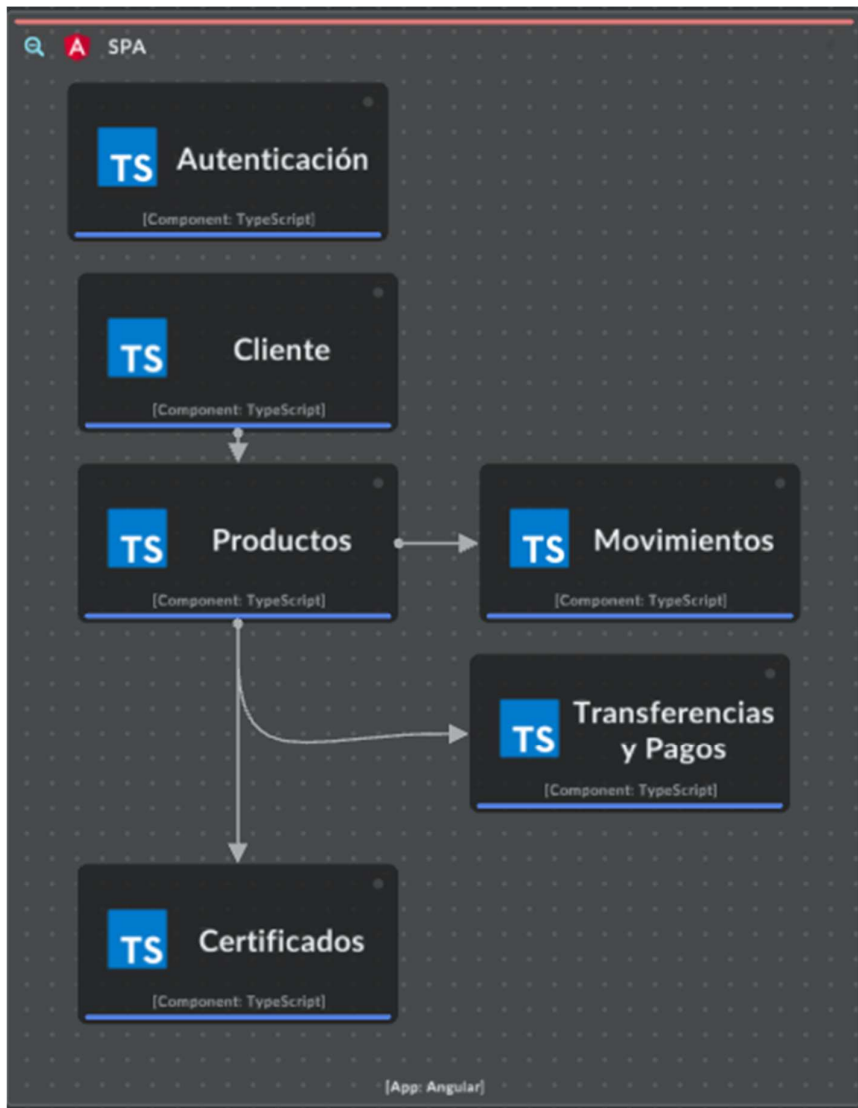
Debido a que la norma exige que los usuarios sean notificados sobre los movimientos realizados, el sistema utilizará sistemas externos o propios de envío de notificaciones, mínimo 2.





Al realizar transferencias o pagos se publicara un mensaje para que los Sistemas de notificación Push, SMS, SMTP envíen las alertas al cliente.

Este sistema contará con 2 aplicaciones en el Front, una SPA



y una Aplicación móvil desarrollada en un Framework multiplataforma. (Mencione 2 opciones y justifique el porqué de su elección)



SPA: Angular

APP: React Native

Otras opciones pueden ser VUE para SPA y Xamarin de Microsoft.

**Angular + React Native** ofrece ventajas sobre **Vue + Xamarin** para una aplicación de banca móvil en términos de seguridad, escalabilidad, rendimiento y soporte.

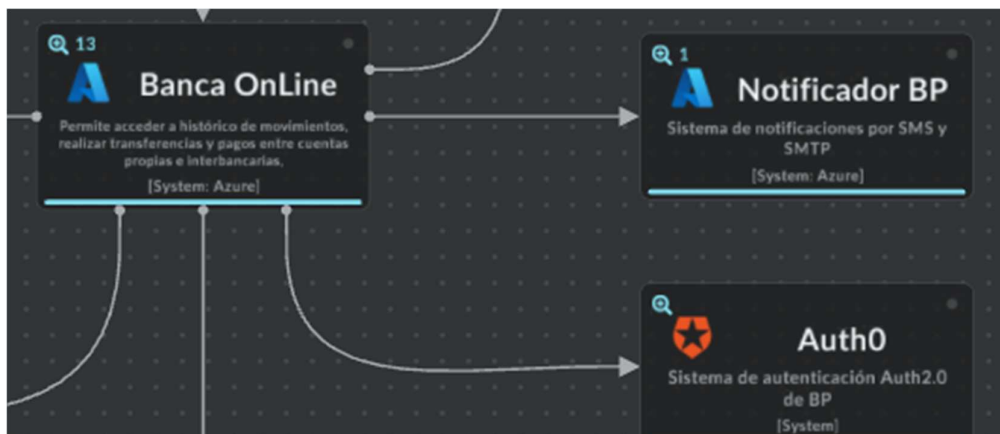
**Angular** es más robusto y seguro que Vue, con mejor estructura para proyectos grandes y protección contra ataques es desarrollado por Google y tiene soporte continuo y parches de seguridad.

**React Native** tiene mejor integración con iOS y Android que Xamarin, proporcionando una mejor experiencia de usuario y rendimiento.

**Seguridad y mantenimiento:** Angular maneja mejor autenticación y protección de datos, y React Native tiene un ecosistema más amplio que facilita la integración y actualizaciones.

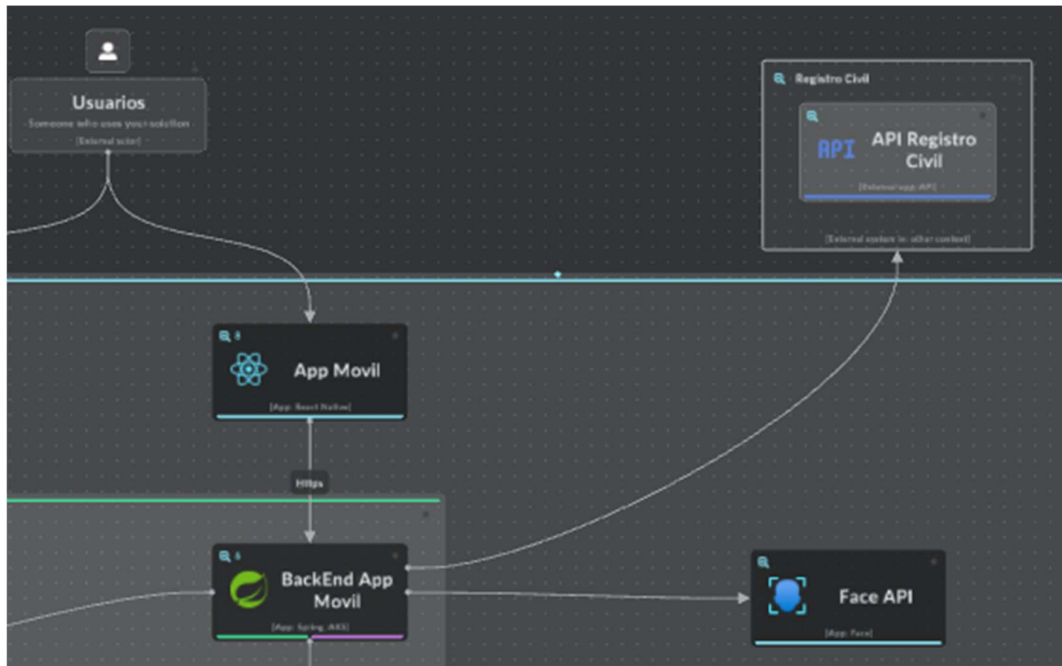
**Compatibilidad y soporte:** React Native recibe mejoras constantes, mientras que Xamarin depende de Microsoft, lo que puede limitar su evolución.

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; sin embargo, debe dar recomendaciones sobre cuál es el mejor flujo de autenticación que se debería usar según el estándar.



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 especifique alguno de los anteriores dentro de su arquitectura, también puede recomendar herramientas de industria que realicen estas tareas y robustezca su aplicación.



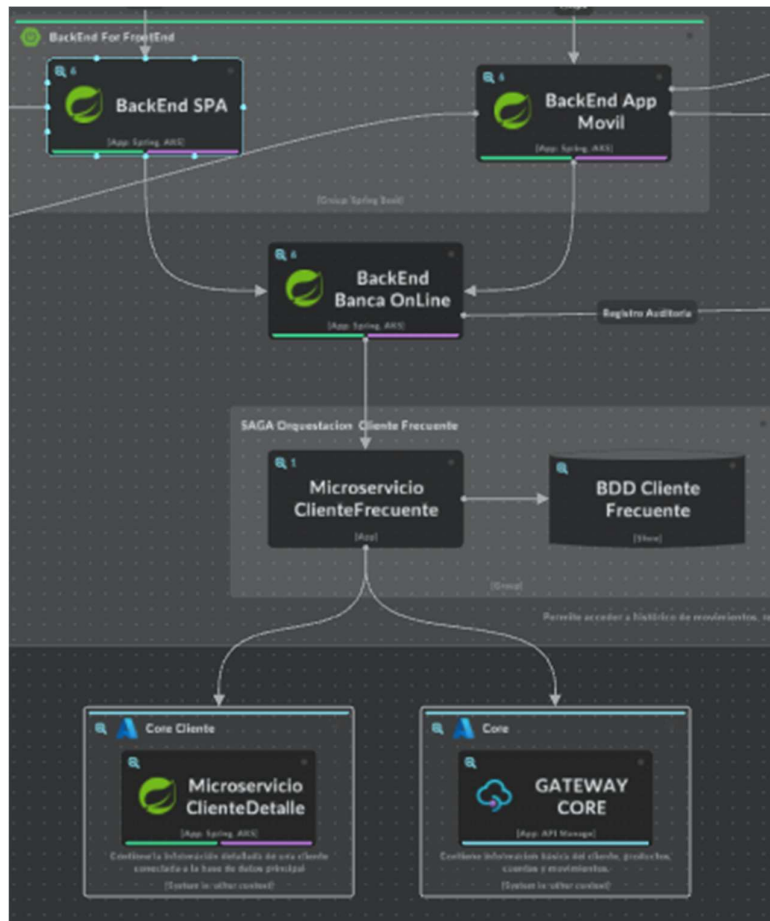
Se utilizará los servicios del registro civil para obtener la foto del cliente mediante su cedula y código dactilar una vez que haya aceptado y dado el consentimiento del uso de su información para la Ley de Protección de Datos personales, solicitaremos una foto que se comparará con el servicio de Azure Face API para validar que se trate de la misma persona y poder dar de alta al cliente en el Onboarding.

El sistema utiliza una base de datos de auditoría que registra todas las acciones del cliente



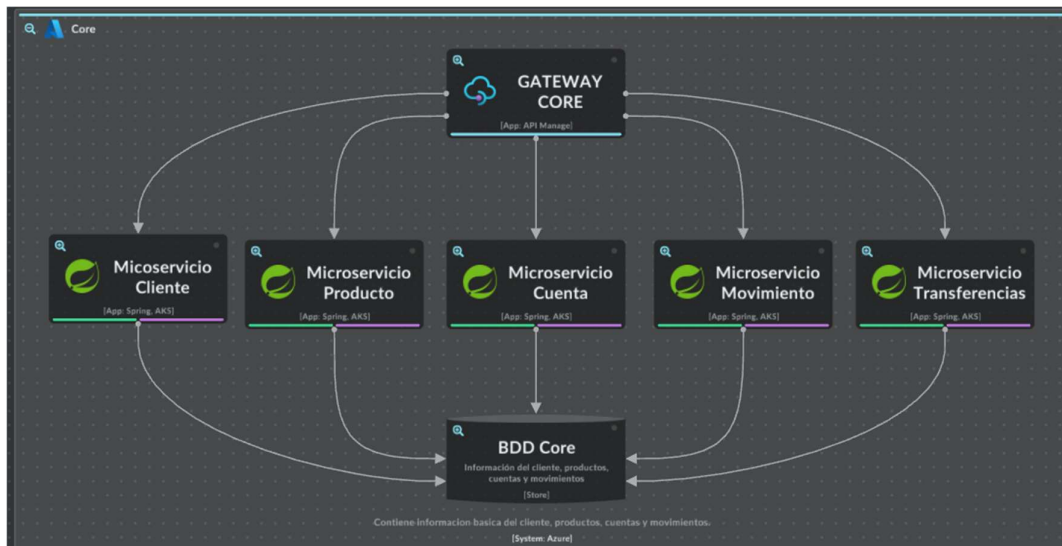
Para este caso vamos a implementar el patrón de publicador/suscriptor mediante Apache Kafka toda acción en la app publicara tópicos para que un subscriber de auditoria los esté escuchando y los vaya procesando sin afectar al rendimiento de la aplicación.

y cuenta con un mecanismo de persistencia de información para clientes frecuentes, para este caso proponga una alternativa basada en patrones de diseño que relacione los componentes que deberían interactuar para conseguir el objetivo.



Existirá un servicio que identificara si el cliente es frecuente y persistirá la información del mismo en una BDD usando el patrón SAGA Orquestación

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 la repuesta de información a sus clientes, es libre de hacerlo.



Para este reto, mencione aquellos elementos normativos que podrían ser importantes a la hora de crear aplicaciones para entidades financieras, Ejemplo ley de datos personales, seguridad etc.

Datos personales.

Intereses, tarifas y comisiones.

ISO 27000

Código Orgánico Monetario y Financiero (COMF):

Define el marco legal para la banca, incluyendo la regulación de la banca electrónica y los sistemas de pago.

Ley Orgánica para Defender los Derechos de los Clientes del Sistema Financiero Nacional:

Protege a los usuarios de la banca, estableciendo normas para la transparencia, la protección de datos y la prevención de abusos.

Resoluciones de la Superintendencia de Bancos:

La Superintendencia de Bancos y Seguros (SB) emite resoluciones que complementan el marco legal y regulan aspectos específicos de la banca electrónica, como la seguridad de los canales electrónicos y la gestión de riesgos.

Resoluciones de la Junta de Política y Regulación Monetaria (JPRM):

La JPRM dicta normativas sobre los medios de pago electrónicos y la actividad de las Fintech.

Aspectos clave de la regulación de la banca electrónica:



#### Seguridad:

Las instituciones financieras deben implementar medidas de seguridad tecnológica y de control para garantizar la integridad, disponibilidad y confidencialidad de los datos de los usuarios en los canales electrónicos.

#### Transparencia:

Las entidades financieras deben informar de manera clara y concisa a los clientes sobre los costos, tarifas, riesgos y beneficios de los productos y servicios financieros ofrecidos a través de canales electrónicos.

#### Protección de datos:

Se debe garantizar la confidencialidad y la protección de los datos personales de los usuarios en los canales electrónicos, cumpliendo con la Ley Orgánica de Protección de Datos Personales.

#### Derechos de los usuarios:

Los usuarios tienen derecho a reclamar, presentar quejas y denunciar irregularidades ante la Superintendencia de Bancos y Seguros o las instancias competentes.

Garantice en su arquitectura, alta disponibilidad (**HA**), tolerancia a fallos, recuperación ante desastres (**DR**), Seguridad y Monitoreo, Excelencia operativa y auto-healing.

### **OnPremise**

#### HA

Para servidores físicos o virtuales, es fundamental desplegar un clúster con balanceo de carga para optimizar la distribución del tráfico y garantizar la disponibilidad.

Si se requieren contenedores OnPremise, BP dispone de OpenShift, que ofrece alta disponibilidad mediante la escalabilidad horizontal de PODs. Esta funcionalidad debe configurarse adecuadamente en los archivos de configuración vinculados a los pipelines de implementación.

#### DR

Banco Pichincha ya cuenta con su DCA implementado, por lo que es necesario desplegar la infraestructura adecuada para la réplica de información. Por ejemplo, los datos y aplicaciones pueden replicarse mediante copias de almacenamiento, mientras que las bases de datos pueden sincronizarse utilizando tecnologías como AlwaysOn o el modelo de Publicador/Distribuidor/Suscriptor.



## Seguridad y Monitoreo, Excelencia operativa y auto-healing

Las aplicaciones OnPremise se conectan a una plataforma de monitoreo a través de agentes encargados de detectar eventos, analizar puertos y realizar búsquedas en logs, presentando los datos en dashboards de supervisión y notificaciones.

El estado de los componentes se evalúa mediante reglas predefinidas, mientras que las acciones correctivas pueden ser ejecutadas de forma automática según reglas establecidas o de manera manual a través de tareas específicas.

## Cloud

### HA

Al implementar esta solución en Azure sobre ASK podemos incluir más seguridad como Microsoft Defender para Kubernetes, incluimos el tema de monitoreo y control y también la excelencia operativa ya que AKS escalara automáticamente de manera horizontal nuestros Microservicios y esto brindara calidad de servicio a nuestros clientes y reducirá el mantenimiento de infraestructura local.

### DR

Dentro de Azure se debe configurar planes con redundancia y replica a otras zonas.

### Tolerancia a fallos:

Al implementar nuestra arquitectura desacoplamos y generamos Microservicios con una única responsabilidad, además implementaremos en los componentes críticos como consulta de clientes y productos un patron de reintentos adicional al cliente frecuente para traer la información, y si servicios como el de movimientos no responde para una cuenta, no parara la funcionalidad de todo el sistema por ello.

## Seguridad y Monitoreo, Excelencia operativa y auto-healing

Al implementar esta solución en Azure sobre ASK podemos incluir más seguridad como Microsoft Defender para Kubernetes, incluimos el tema de monitoreo y control y también la excelencia operativa ya que AKS escalara automáticamente de manera horizontal nuestros Microservicios y esto brindara calidad de servicio a nuestros clientes y reducirá el mantenimiento de infraestructura local.

Para el Auto-healing vamos a aprovechar la capacidad de Azure dentro del Azure App Service donde configuraremos las reglas y las acciones que Azure auto healing services.

Sí lo considera necesario, su arquitectura puede contener elementos de infraestructura en nube como Azure o AWS, garantice baja latencia en sus servicios, cuenta con presupuesto para esto.

Se implemento microservicios sobre Azure que se despliegan sobre AKS lo que brinda resiliencia, escalabilidad, supervisión y monitoreo

En lo posible plantee una arquitectura desacoplada con elementos y reusables y cohesionados para otros componentes que puedan adicionarse en el futuro.

Toda la arquitectura se implemento con Microservicios respetando los principios SOLID, para que cumplan una sola función y se encuentren desacoplados siendo reutilizables.



El modelo debe ser desarrollado bajo **c4**(Modelo de Contexto, Modelo de aplicación o contenedor y Componentes), describa hasta el modelo de componentes, la infraestructura la puede modelar como usted lo considere usando la herramienta de su preferencia.

Link del modelo C4 en IcePanel: <https://s.icepanel.io/MJn9sBcJNj3Wp6/LRsb>

Link Github: <https://github.com/majurado/TestArquitectura>

**¡Éxitos!**

# Default domain domain

## Alexander Jurado's landscape

Generated by [IcePanel](#) at 2025-05-22 13:16:00

## Table of contents

- [1. Context diagrams](#)
  - [1.1. Banca OnLine - Contexto](#)
- [2. App diagrams](#)
  - [2.1. Banca OnLine App Diagram](#)
  - [2.2. Core App Diagram](#)
  - [2.3. Core Cliente App Diagram](#)
  - [2.4. Notificador BP App Diagram](#)
  - [2.5. Notificador Push BP App Diagram](#)
  - [2.6. Registro Civil App Diagram](#)
- [3. App diagram flows](#)
  - [3.1. New customer flow](#)
- [4. Component diagrams](#)
  - [4.1. App Movil Component Diagram](#)
  - [4.2. BackEnd App Movil Component Diagram](#)
  - [4.3. BackEnd Banca OnLine Component Diagram](#)
  - [4.4. BackEnd SPA Component Diagram](#)
  - [4.5. Microservicio ClienteFrecuente Component Diagram](#)
  - [4.6. MicroservicioNotificaciones Component Diagram](#)
  - [4.7. SPA Component Diagram](#)
  - [4.8. Subscriptor Auditoria Component Diagram](#)
  - [4.9. Subscriptor Notificacion Component Diagram](#)
- [5. Actors](#)
  - [5.1. Usuarios](#)
- [6. Groups](#)

- [6.1. BackEnd For FrontEnd](#)
- [6.2. Eventos Auditoria](#)
- [6.3. SAGA Orquestacion Cliente Frecuente](#)
- [7. Systems](#)
  - [7.1. Auth0](#)
  - [7.2. Banca OnLine](#)
  - [7.3. Core](#)
  - [7.4. Core Cliente](#)
  - [7.5. Notificador BP](#)
  - [7.6. Notificador Push BP](#)
  - [7.7. Registro Civil](#)
- [8. Apps](#)
  - [8.1. API Registro Civil](#)
  - [8.2. App Movil](#)
  - [8.3. BackEnd App Movil](#)
  - [8.4. BackEnd Banca OnLine](#)
  - [8.5. BackEnd SPA](#)
  - [8.6. core.bp](#)
  - [8.7. Face API](#)
  - [8.8. GATEWAY CORE](#)
  - [8.9. Micoservicio Cliente](#)
  - [8.10. Micoservicio ClienteDetalle](#)
  - [8.11. Micoservicio ClienteFrecuente](#)
  - [8.12. Micoservicio Cuenta](#)
  - [8.13. Micoservicio Movimiento](#)
  - [8.14. Micoservicio Notificaciones Push](#)
  - [8.15. Micoservicio Producto](#)
  - [8.16. Micoservicio Transferencias](#)
  - [8.17. MicoservicioNotificaciones](#)
  - [8.18. SPA](#)
  - [8.19. Subscriptor Auditoria](#)

- [8.20. Subscriptor Notificacion](#)
- [9. Stores](#)
  - [9.1. Store](#)
  - [9.2. BDD Auditoria](#)
  - [9.3. Bdd Cliente](#)
  - [9.4. BDD Cliente Frecuente](#)
  - [9.5. BDD Core](#)
- [10. Components](#)
  - [10.1. Auditoria Controller](#)
  - [10.2. Autenticación](#)
  - [10.3. Autenticacion BP Controller](#)
  - [10.4. Autenticacion Controller](#)
  - [10.5. Autenticacion Movil](#)
  - [10.6. Autenticacion Movil Controller](#)
  - [10.7. Autenticación Movil Face ID](#)
  - [10.8. Autenticacion Movil Huella](#)
  - [10.9. Certificados](#)
  - [10.10. Certificados BP Controller](#)
  - [10.11. Certificados Controller](#)
  - [10.12. Certificados Movil](#)
  - [10.13. Certificados Movil Controller](#)
  - [10.14. Cliente](#)
  - [10.15. Cliente Controller](#)
  - [10.16. Cliente Frecuente Controller](#)
  - [10.17. Cliente Movil](#)
  - [10.18. Cliente Movil Controller](#)
  - [10.19. Clientes BP Controller](#)
  - [10.20. Movimiento Controller](#)
  - [10.21. Movimiento Movil Controller](#)
  - [10.22. Movimientos](#)
  - [10.23. Movimientos BP Controller](#)

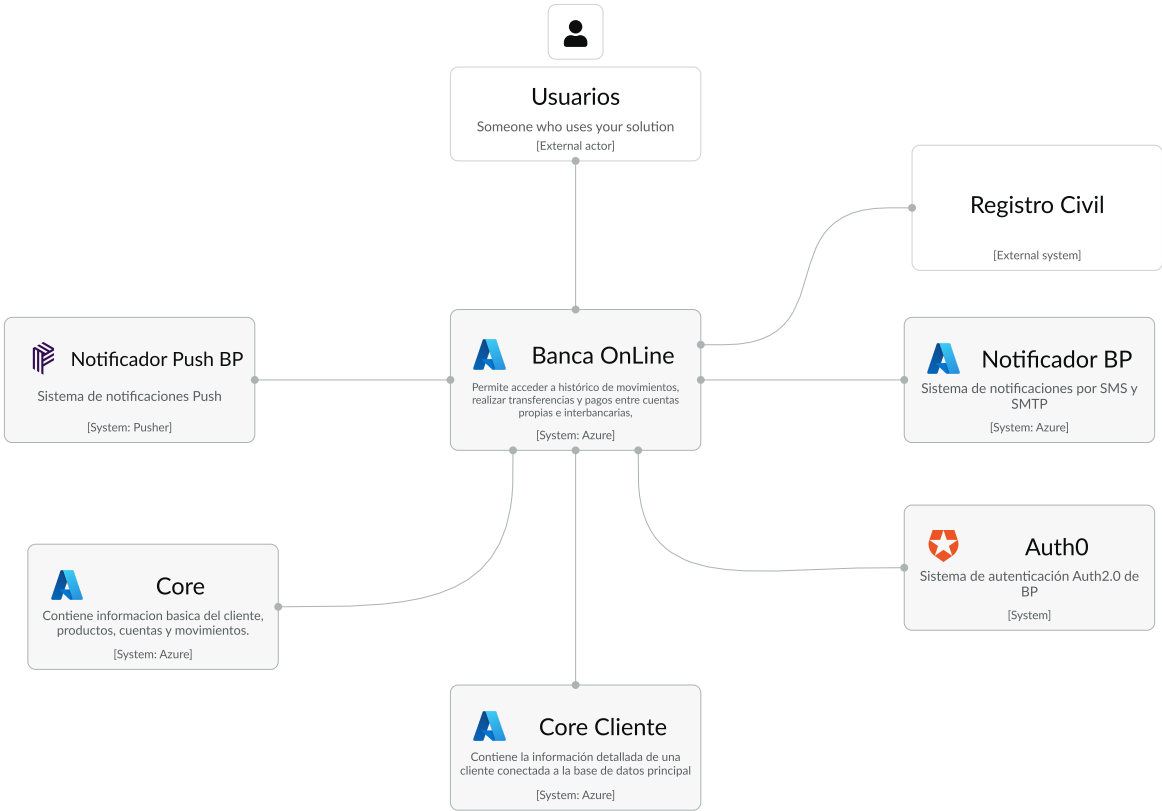
- [10.24. Movimientos Movil](#)
- [10.25. Notificacion Controller](#)
- [10.26. Onboarding BP Controller](#)
- [10.27. Onboarding Movil Controller](#)
- [10.28. Producto BP Controller](#)
- [10.29. Producto Controller](#)
- [10.30. Producto Movil Controller](#)
- [10.31. Productos](#)
- [10.32. Productos Movil](#)
- [10.33. SMS Controller](#)
- [10.34. SMTP Controller](#)
- [10.35. Transferencias y Pagos](#)
- [10.36. Transferencias y Pagos BP Controller](#)
- [10.37. Transferencias y Pagos Controller](#)
- [10.38. Transferencias y Pagos Movil](#)
- [10.39. Transferencias y Pagos Movil Controller](#)



# 1. Context diagrams

## 1.1. Banca OnLine - Contexto

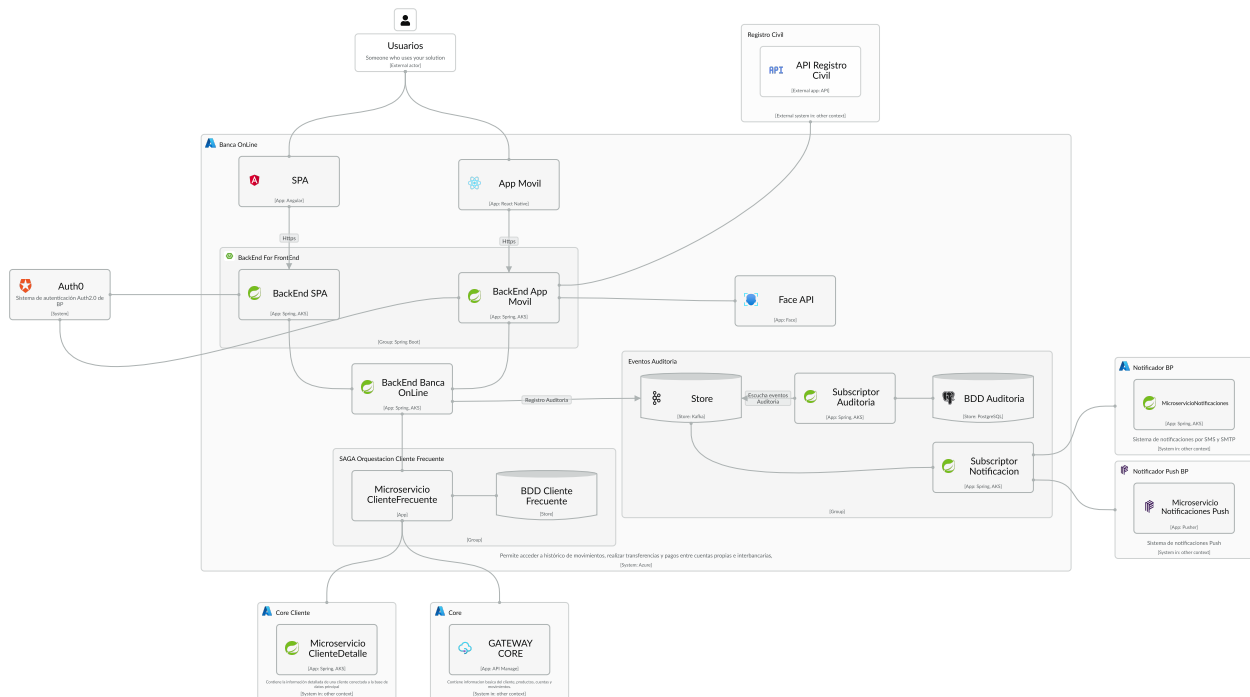
Context diagram in the Default domain domain.



## 2. App diagrams

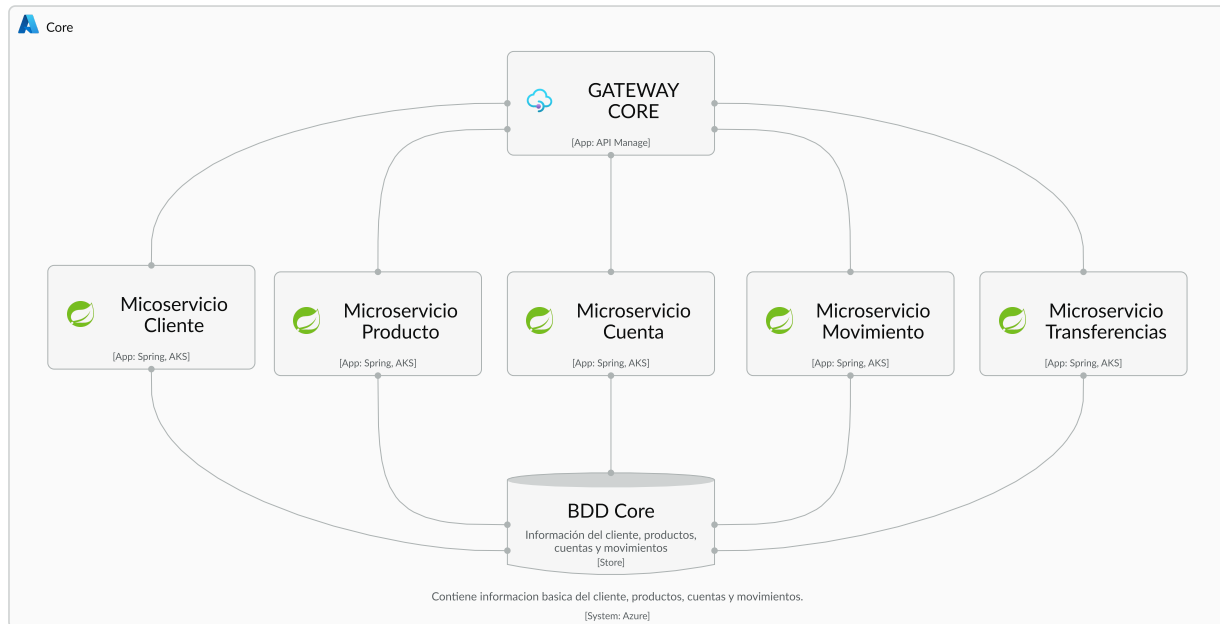
### 2.1. Banca OnLine App Diagram

Banca OnLine app diagram in the Default domain domain.



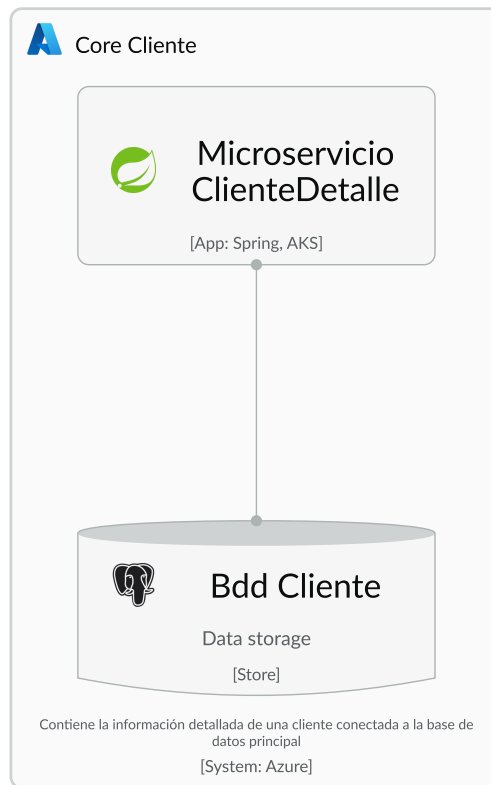
## 2.2. Core App Diagram

Core app diagram in the Default domain domain.



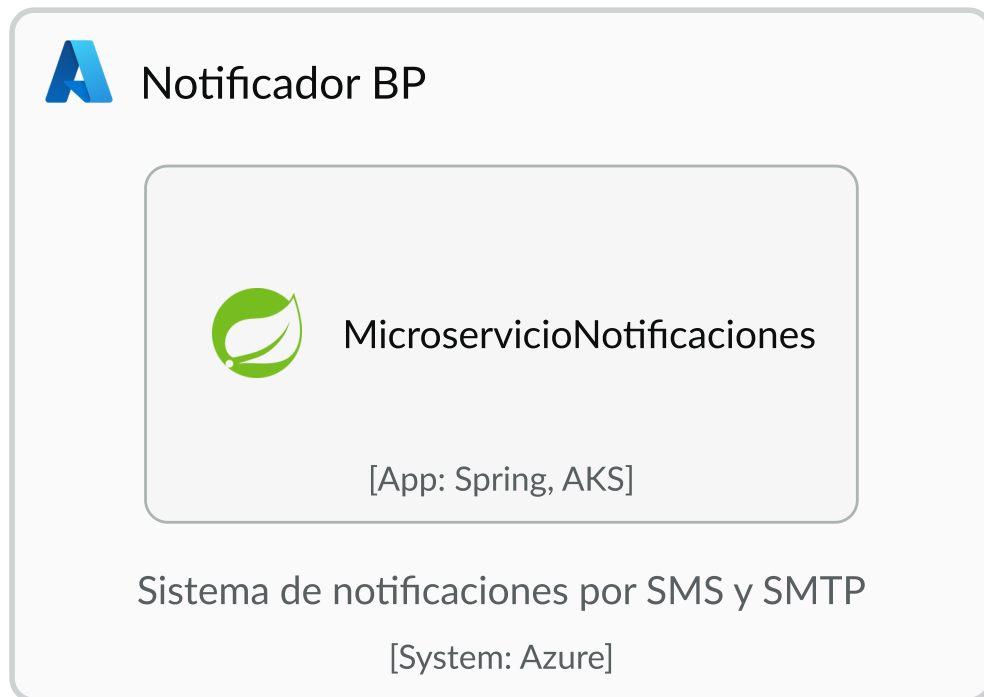
## 2.3. Core Cliente App Diagram

Core Cliente app diagram in the Default domain domain.



## 2.4. Notificador BP App Diagram

Notificador BP app diagram in the Default domain domain.



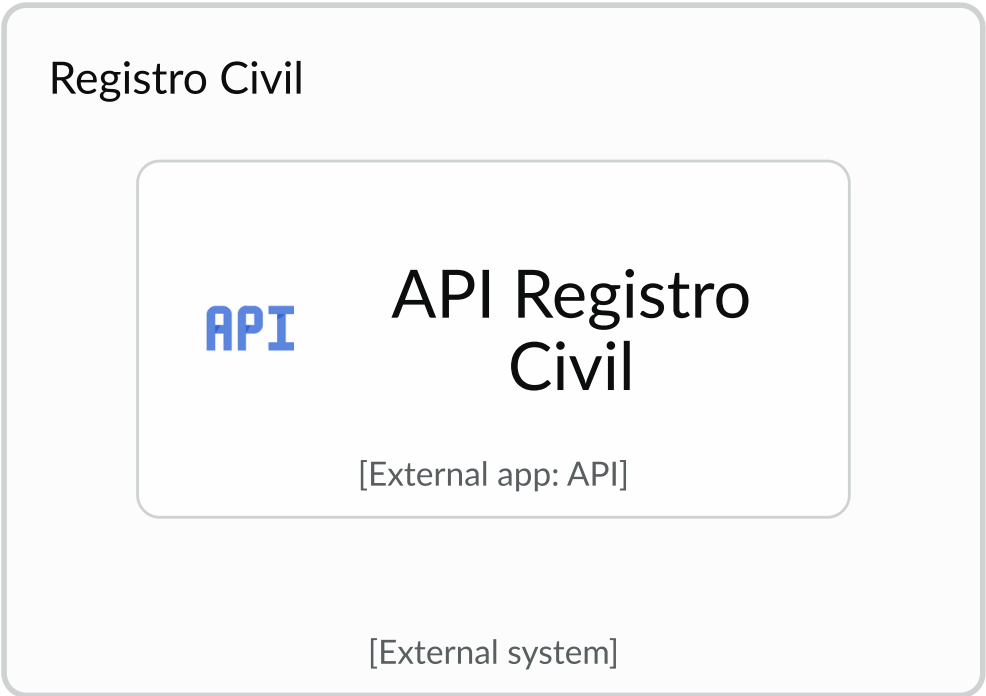
## 2.5. Notificador Push BP App Diagram

Notificador Push BP app diagram in the Default domain domain.



2.6. Registro Civil App Diagram

Registro Civil app diagram in the Default domain domain.



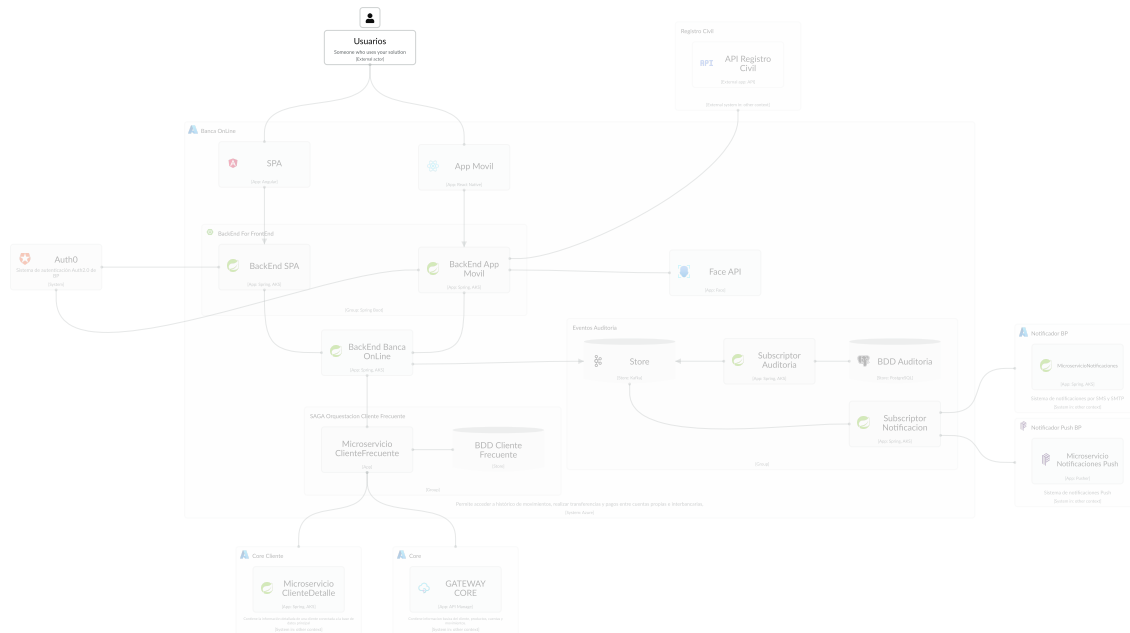


### 3. App diagram flows

#### 3.1. New customer flow

App diagram flow shown on the Banca OnLine App Diagram diagram in the Default domain domain.

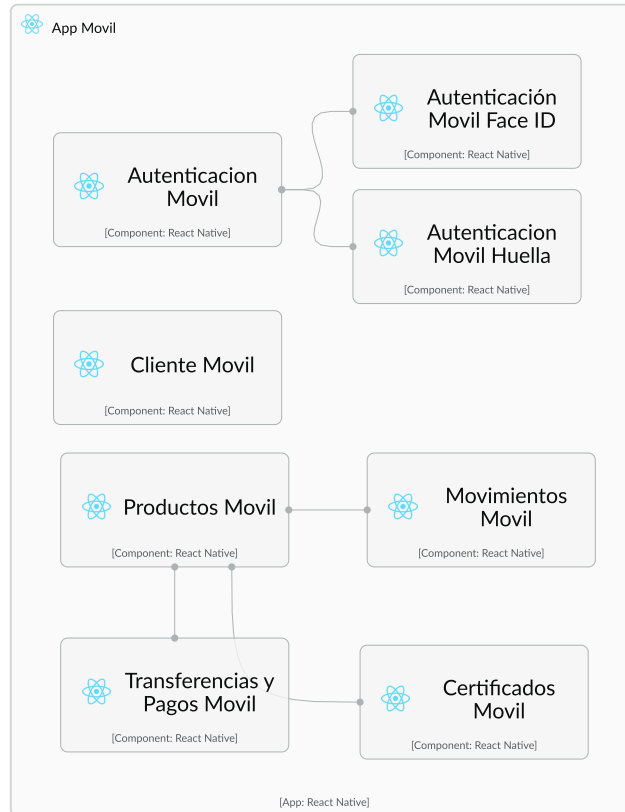
Steps:



## 4. Component diagrams

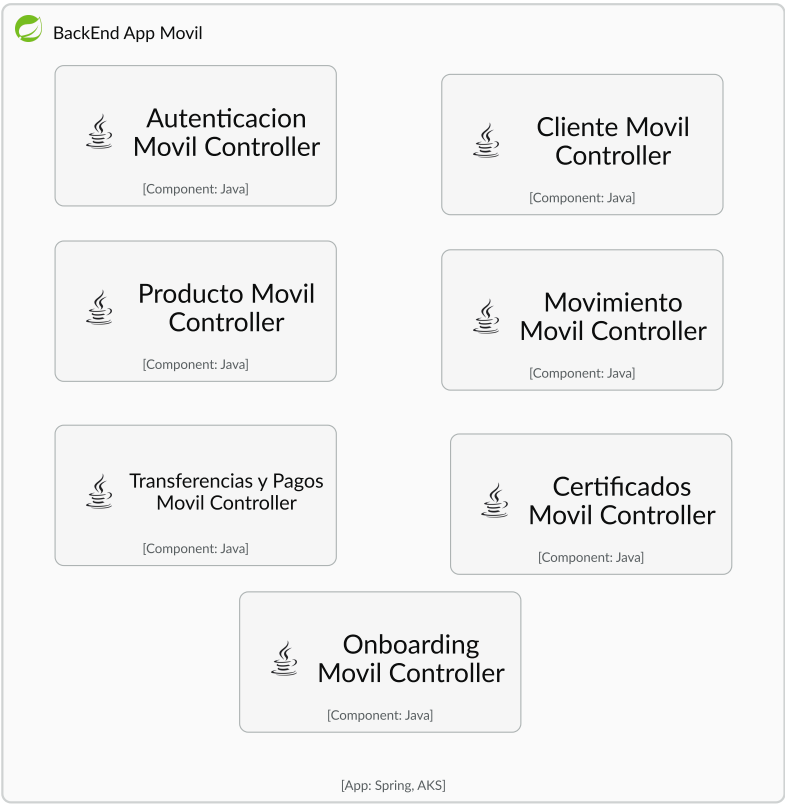
### 4.1. App Movil Component Diagram

App Movil component diagram in the Default domain.



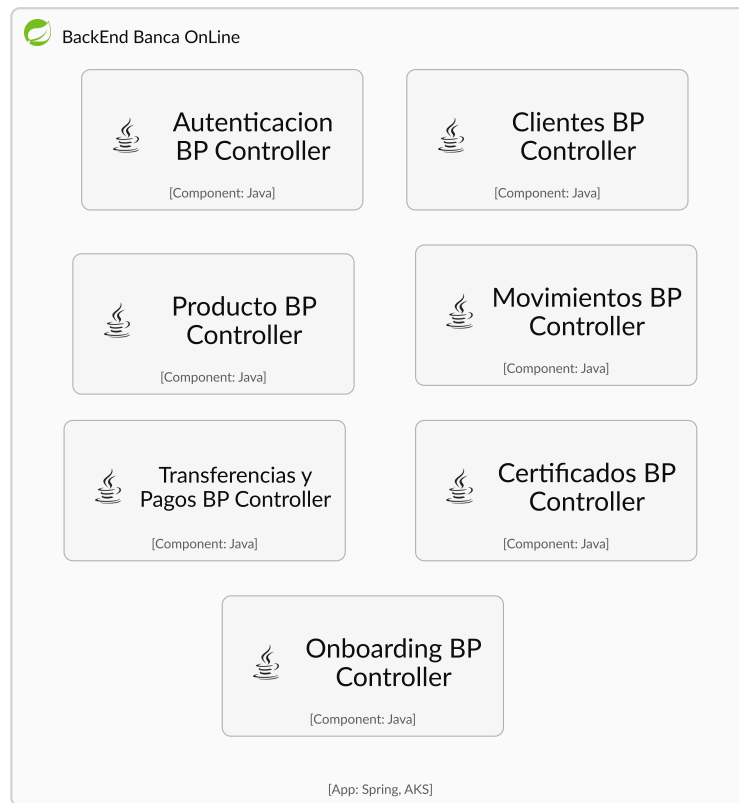
## 4.2. BackEnd App Movil Component Diagram

BackEnd App Movil component diagram in the Default domain domain.



### 4.3. BackEnd Banca OnLine Component Diagram

BackEnd Banca OnLine component diagram in the Default domain domain.



### 4.4. BackEnd SPA Component Diagram

BackEnd SPA component diagram in the Default domain domain.



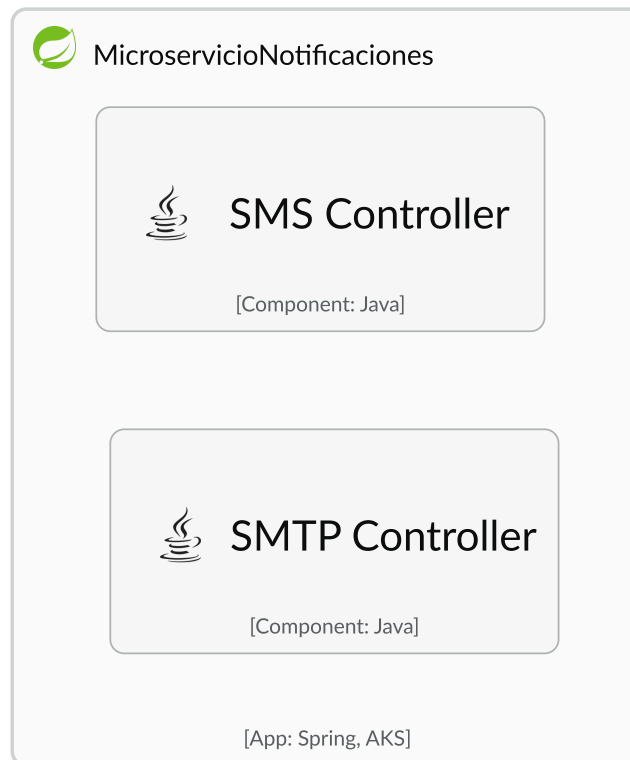
## 4.5. Microservicio ClienteFrecuente Component Diagram

Microservicio ClienteFrecuente component diagram in the Default domain domain.



## 4.6. MicroservicioNotificaciones Component Diagram

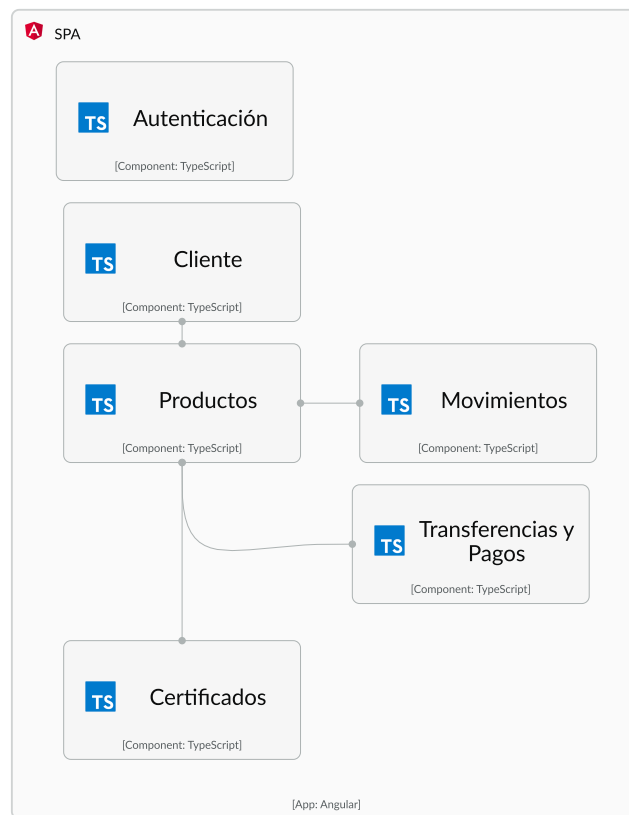
MicroservicioNotificaciones component diagram in the Default domain domain.





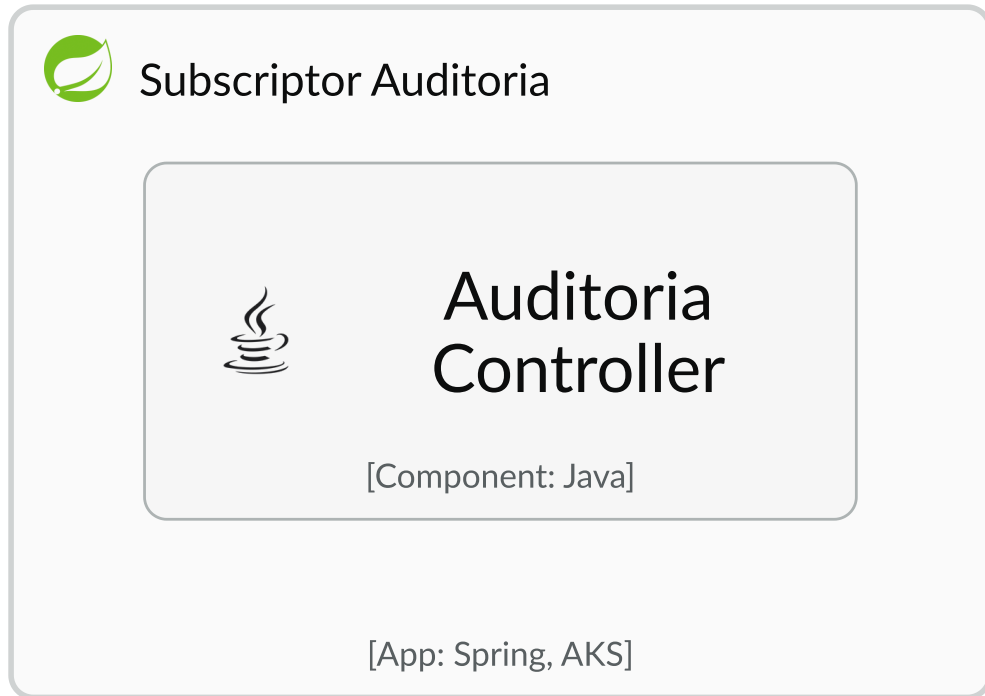
## 4.7. SPA Component Diagram

SPA component diagram in the Default domain domain.



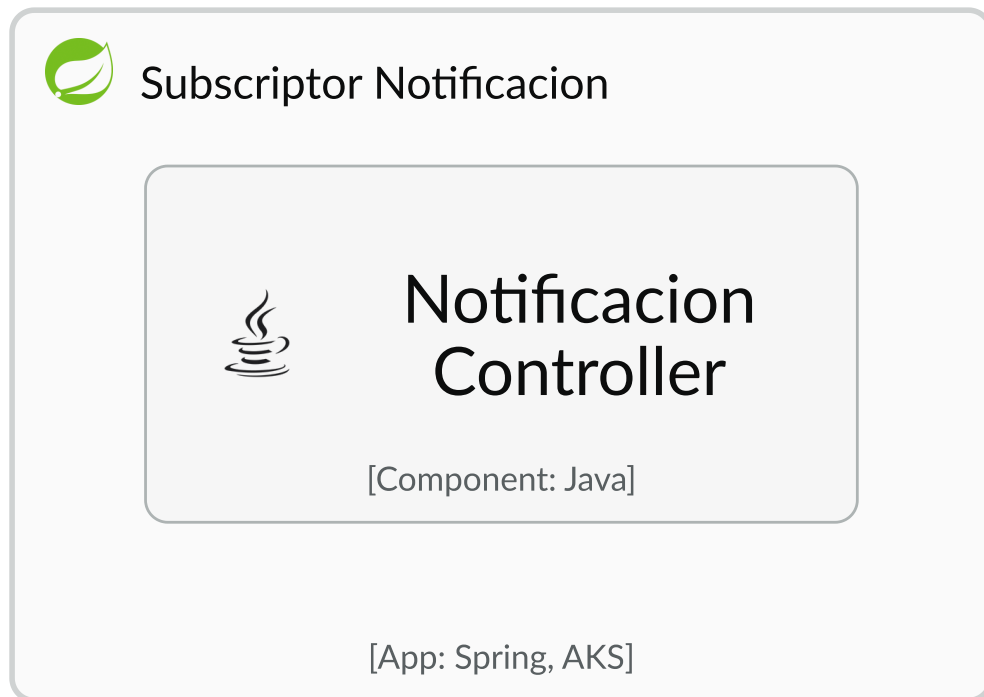
## 4.8. Subscriptor Auditoria Component Diagram

Subscriptor Auditoria component diagram in the Default domain domain.



## 4.9. Subscriptor Notificacion Component Diagram

Subscriptor Notificacion component diagram in the Default domain domain.



## 5. Actors

### 5.1. Usuarios

External actor in the Default domain domain.

Status: live

Someone who uses your solution

**In diagrams:**

- [Banca OnLine - Contexto](#)
- [Banca OnLine App Diagram](#)

**In diagrams:**

- [New customer flow](#)

## 6. Groups

### 6.1. BackEnd For FrontEnd

Internal group in the Default domain domain.

Status: live

**Technologies:**

- [Spring Boot](#)

**In diagrams:**

- [Banca OnLine App Diagram](#)

### 6.2. Eventos Auditoria

Internal group in the Default domain domain.

Status: live

**In diagrams:**

- [Banca OnLine App Diagram](#)

### 6.3. SAGA Orquestacion Cliente Frecuente

Internal group in the Default domain domain.

Status: live

**In diagrams:**

- [Banca OnLine App Diagram](#)

## 7. Systems

### 7.1. Auth0

Internal system in the Default domain domain.

Status: live

Sistema de autenticación Auth2.0 de BP

#### Technologies:

- [Auth0](#)

#### In diagrams:

- [Banca OnLine - Contexto](#)
- [Banca OnLine App Diagram](#)

### 7.2. Banca OnLine

Internal system in the Default domain domain.

Status: live

Permite acceder a histórico de movimientos, realizar transferencias y pagos entre cuentas propias e interbancarias,

#### Technologies:

- [Microsoft Azure](#)

#### In diagrams:

- [Banca OnLine - Contexto](#)
- [Banca OnLine App Diagram](#)

A brief description of your system

### 7.3. Core

Internal system in the Default domain domain.

Status: live

Contiene informacion basica del cliente, productos, cuentas y movimientos.

#### Technologies:

- [Microsoft Azure](#)

#### In diagrams:

- [Banca OnLine - Contexto](#)

- [Banca OnLine App Diagram](#)
- [Core App Diagram](#)

## 7.4. Core Cliente

Internal system in the Default domain domain.

Status: live

Contiene la información detallada de una cliente conectada a la base de datos principal

### Technologies:

- [Microsoft Azure](#)

### In diagrams:

- [Banca OnLine - Contexto](#)
- [Banca OnLine App Diagram](#)
- [Core Cliente App Diagram](#)

## 7.5. Notificador BP

Internal system in the Default domain domain.

Status: live

Sistema de notificaciones por SMS y SMTP

### Technologies:

- [Microsoft Azure](#)

### In diagrams:

- [Banca OnLine - Contexto](#)
- [Banca OnLine App Diagram](#)
- [Notificador BP App Diagram](#)

## 7.6. Notificador Push BP

Internal system in the Default domain domain.

Status: live

Sistema de notificaciones Push

### Technologies:

- [Pusher](#)

### In diagrams:

- [Banca OnLine - Contexto](#)
- [Banca OnLine App Diagram](#)
- [Notificador Push BP App Diagram](#)

## 7.7. Registro Civil

External system in the Default domain domain.

Status: live

**In diagrams:**

- [Banca OnLine - Contexto](#)
- [Banca OnLine App Diagram](#)
- [Registro Civil App Diagram](#)

## 8. Apps

### 8.1. API Registro Civil

External app in the Default domain domain.

Status: live

**Technologies:**

- [API](#)

**In diagrams:**

- [Banca OnLine App Diagram](#)
- [Registro Civil App Diagram](#)

### 8.2. App Movil

Internal app in the Default domain domain.

Status: live

**Technologies:**

- [React Native](#)

**In diagrams:**

- [App Movil Component Diagram](#)
- [Banca OnLine App Diagram](#)



### 8.3. BackEnd App Movil

Internal app in the Default domain domain.

Status: live

**Technologies:**

- [Spring](#)
- [Azure Kubernetes Service \(AKS\)](#)

**In groups:**

- [BackEnd For FrontEnd](#)

**In diagrams:**

- [BackEnd App Movil Component Diagram](#)
- [Banca OnLine App Diagram](#)

### 8.4. BackEnd Banca OnLine

Internal app in the Default domain domain.

Status: live

**Technologies:**

- [Spring](#)
- [Azure Kubernetes Service \(AKS\)](#)

**In diagrams:**

- [BackEnd Banca OnLine Component Diagram](#)
- [Banca OnLine App Diagram](#)

### 8.5. BackEnd SPA

Internal app in the Default domain domain.

Status: live

**Technologies:**

- [Spring](#)
- [Azure Kubernetes Service \(AKS\)](#)

**In groups:**

- [BackEnd For FrontEnd](#)

**In diagrams:**

- [BackEnd SPA Component Diagram](#)
- [Banca OnLine App Diagram](#)

## 8.6. core.bp

Internal app in the Default domain domain.

Status: live

Backend REST API

**In diagrams:**

- [New customer flow](#)

## 8.7. Face API

Internal app in the Default domain domain.

Status: live

**Technologies:**

- [Azure Face](#)

**In diagrams:**

- [Banca OnLine App Diagram](#)

## 8.8. GATEWAY CORE

Internal app in the Default domain domain.

Status: live

**Technologies:**

- [API Management Services](#)

**In diagrams:**

- [Banca OnLine App Diagram](#)
- [Core App Diagram](#)

## 8.9. Micoservicio Cliente

Internal app in the Default domain domain.

Status: live

**Technologies:**

- [Spring](#)

- [Azure Kubernetes Service \(AKS\)](#)

**In diagrams:**

- [Core App Diagram](#)

## 8.10. Microservicio ClienteDetalle

Internal app in the Default domain domain.

Status: live

**Technologies:**

- [Spring](#)
- [Azure Kubernetes Service \(AKS\)](#)

**In diagrams:**

- [Banca OnLine App Diagram](#)
- [Core Cliente App Diagram](#)

## 8.11. Microservicio ClienteFrecuente

Internal app in the Default domain domain.

Status: live

**In groups:**

- [SAGA Orquestacion Cliente Frecuente](#)

**In diagrams:**

- [Banca OnLine App Diagram](#)
- [Microservicio ClienteFrecuente Component Diagram](#)

## 8.12. Microservicio Cuenta

Internal app in the Default domain domain.

Status: live

**Technologies:**

- [Spring](#)
- [Azure Kubernetes Service \(AKS\)](#)

**In diagrams:**

- [Core App Diagram](#)

### 8.13. Microservicio Movimiento

Internal app in the Default domain domain.

Status: live

**Technologies:**

- [Spring](#)
- [Azure Kubernetes Service \(AKS\)](#)

**In diagrams:**

- [Core App Diagram](#)

### 8.14. Microservicio Notificaciones Push

Internal app in the Default domain domain.

Status: live

**Technologies:**

- [Pusher](#)

**In diagrams:**

- [Banca OnLine App Diagram](#)
- [Notificador Push BP App Diagram](#)

### 8.15. Microservicio Producto

Internal app in the Default domain domain.

Status: live

**Technologies:**

- [Spring](#)
- [Azure Kubernetes Service \(AKS\)](#)

**In diagrams:**

- [Core App Diagram](#)

### 8.16. Microservicio Transferencias

Internal app in the Default domain domain.

Status: live

**Technologies:**

- [Spring](#)
- [Azure Kubernetes Service \(AKS\)](#)

In diagrams:

- [Core App Diagram](#)

## 8.17. MicroservicioNotificaciones

Internal app in the Default domain domain.

Status: live

Technologies:

- [Spring](#)
- [Azure Kubernetes Service \(AKS\)](#)

In diagrams:

- [Banca OnLine App Diagram](#)
- [MicroservicioNotificaciones Component Diagram](#)
- [Notificador BP App Diagram](#)

## 8.18. SPA

Internal app in the Default domain domain.

Status: live

Technologies:

- [Angular](#)

In diagrams:

- [Banca OnLine App Diagram](#)
- [SPA Component Diagram](#)

## 8.19. Subscriptor Auditoria

Internal app in the Default domain domain.

Status: live

Technologies:

- [Spring](#)
- [Azure Kubernetes Service \(AKS\)](#)

**In groups:**

- [Eventos Auditoria](#)

**In diagrams:**

- [Banca OnLine App Diagram](#)
- [Subscriber Auditoria Component Diagram](#)

## 8.20. Subscriber Notification

Internal app in the Default domain domain.

Status: live

**Technologies:**

- [Spring](#)
- [Azure Kubernetes Service \(AKS\)](#)

**In groups:**

- [Eventos Auditoria](#)

**In diagrams:**

- [Banca OnLine App Diagram](#)
- [Subscriber Notification Component Diagram](#)

## 9. Stores

### 9.1. Store

Internal store in the Default domain domain.

Status: live

**Technologies:**

- [Apache Kafka](#)

**In groups:**

- [Eventos Auditoria](#)

**In diagrams:**

- [Banca OnLine App Diagram](#)

### 9.2. BDD Auditoria

Internal store in the Default domain domain.

Status: live

**Technologies:**

- [PostgreSQL](#)

**In groups:**

- [Eventos Auditoria](#)

**In diagrams:**

- [Banca OnLine App Diagram](#)

### 9.3. Bdd Cliente

Internal store in the Default domain domain.

Status: live

Data storage

**In diagrams:**

- [Core Cliente App Diagram](#)

**In diagrams:**

- [New customer flow](#)

### 9.4. BDD Cliente Frecuente

Internal store in the Default domain domain.

Status: live

**In groups:**

- [SAGA Orquestacion Cliente Frecuente](#)

**In diagrams:**

- [Banca OnLine App Diagram](#)

### 9.5. BDD Core

Internal store in the Default domain domain.

Status: live

Información del cliente, productos, cuentas y movimientos

**In diagrams:**

- [Core App Diagram](#)

## 10. Components

### 10.1. Auditoria Controller

Internal component in the Default domain domain.

Status: live

#### Technologies:

- [Java](#)

#### In diagrams:

- [Subscriber Auditoria Component Diagram](#)

### 10.2. Autenticación

Internal component in the Default domain domain.

Status: live

#### Technologies:

- [TypeScript](#)

#### In diagrams:

- [SPA Component Diagram](#)

### 10.3. Autenticacion BP Controller

Internal component in the Default domain domain.

Status: live

#### Technologies:

- [Java](#)

#### In diagrams:

- [BackEnd Banca OnLine Component Diagram](#)

### 10.4. Autenticacion Controller

Internal component in the Default domain domain.

Status: live

#### Technologies:

- [Java](#)

#### In diagrams:

- [BackEnd SPA Component Diagram](#)

### 10.5. Autenticacion Movil



Internal component in the Default domain domain.

Status: live

**Technologies:**

- [React Native](#)

**In diagrams:**

- [App Movil Component Diagram](#)

## 10.6. Autenticacion Movil Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [BackEnd App Movil Component Diagram](#)

## 10.7. Autenticación Movil Face ID

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [React Native](#)

**In diagrams:**

- [App Movil Component Diagram](#)

## 10.8. Autenticacion Movil Huella

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [React Native](#)

**In diagrams:**

- [App Movil Component Diagram](#)

## 10.9. Certificados

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [TypeScript](#)

**In diagrams:**

- [SPA Component Diagram](#)

## 10.10. Certificados BP Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [BackEnd Banca OnLine Component Diagram](#)

## 10.11. Certificados Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [BackEnd SPA Component Diagram](#)

## 10.12. Certificados Movil

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [React Native](#)

**In diagrams:**

- [App Movil Component Diagram](#)

## 10.13. Certificados Movil Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [BackEnd App Movil Component Diagram](#)

## 10.14. Cliente

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [TypeScript](#)

**In diagrams:**

- [SPA Component Diagram](#)

## 10.15. Cliente Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [BackEnd SPA Component Diagram](#)

## 10.16. Cliente Frecuente Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [Microservicio ClienteFrecuente Component Diagram](#)

## 10.17. Cliente Movil

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [React Native](#)

**In diagrams:**

- [App Movil Component Diagram](#)

## 10.18. Cliente Movil Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [BackEnd App Movil Component Diagram](#)

## 10.19. Clientes BP Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [BackEnd Banca OnLine Component Diagram](#)

## 10.20. Movimiento Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [BackEnd SPA Component Diagram](#)

## 10.21. Movimiento Movil Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [BackEnd App Movil Component Diagram](#)

## 10.22. Movimientos

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [TypeScript](#)

**In diagrams:**

- [SPA Component Diagram](#)

## 10.23. Movimientos BP Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [BackEnd Banca OnLine Component Diagram](#)

## 10.24. Movimientos Movil

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [React Native](#)

**In diagrams:**

- [App Movil Component Diagram](#)

## 10.25. Notificacion Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [Subscriber Notificacion Component Diagram](#)

## 10.26. Onboarding BP Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [BackEnd Banca OnLine Component Diagram](#)

## 10.27. Onboarding Movil Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [BackEnd App Movil Component Diagram](#)

## 10.28. Producto BP Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [BackEnd Banca OnLine Component Diagram](#)

## 10.29. Producto Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [BackEnd SPA Component Diagram](#)

### 10.30. Producto Movil Controller

Internal component in the Default domain domain.

Status: live

#### Technologies:

- [Java](#)

#### In diagrams:

- [BackEnd App Movil Component Diagram](#)

### 10.31. Productos

Internal component in the Default domain domain.

Status: live

#### Technologies:

- [TypeScript](#)

#### In diagrams:

- [SPA Component Diagram](#)

### 10.32. Productos Movil

Internal component in the Default domain domain.

Status: live

#### Technologies:

- [React Native](#)

#### In diagrams:

- [App Movil Component Diagram](#)

### 10.33. SMS Controller

Internal component in the Default domain domain.

Status: live

#### Technologies:

- [Java](#)

#### In diagrams:

- [MicroservicioNotificaciones Component Diagram](#)

### 10.34. SMTP Controller

Internal component in the Default domain domain.

Status: live

#### Technologies:

- [Java](#)

#### In diagrams:

- [MicroservicioNotificaciones Component Diagram](#)

### 10.35. Transferencias y Pagos

Internal component in the Default domain domain.

Status: live

#### Technologies:

- [TypeScript](#)

#### In diagrams:

- [SPA Component Diagram](#)

### 10.36. Transferencias y Pagos BP Controller

Internal component in the Default domain domain.

Status: live

#### Technologies:

- [Java](#)

#### In diagrams:

- [BackEnd Banca OnLine Component Diagram](#)

### 10.37. Transferencias y Pagos Controller

Internal component in the Default domain domain.

Status: live

#### Technologies:

- [Java](#)

#### In diagrams:

- [BackEnd SPA Component Diagram](#)

### 10.38. Transferencias y Pagos Movil



Internal component in the Default domain domain.

Status: live

**Technologies:**

- [React Native](#)

**In diagrams:**

- [App Movil Component Diagram](#)

## 10.39. Transferencias y Pagos Movil Controller

Internal component in the Default domain domain.

Status: live

**Technologies:**

- [Java](#)

**In diagrams:**

- [BackEnd App Movil Component Diagram](#)