



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Sistemas

**Migración del backend del canal agente de una entidad
financiera hacia una arquitectura de microservicios**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Erick Herminio CHIPANA RAMOS

ASESOR

Luzmila Elisa PRÓ CONCEPCIÓN

Lima, Perú

2021



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Chipana, E. (2021). *Migración del backend del canal agente de una entidad financiera hacia una arquitectura de microservicios*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	ERICK HERMINIO CHIPANA RAMOS
Tipo de documento de identidad	DNI
Número de documento de identidad	70439677
URL de ORCID	https://orcid.org/0000-0002-9389-591X
Datos de asesor	
Nombres y apellidos	Luzmila Elisa Pró Concepción
Tipo de documento de identidad	DNI
Número de documento de identidad	08862360
URL de ORCID	https://orcid.org/0000-0003-0622-1173
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Pablo Jesús Romero Naupari
Tipo de documento	DNI
Número de documento de identidad	06182185
Miembro del jurado 1	
Nombres y apellidos	Carlos Enrique Yáñez Durán
Tipo de documento	DNI
Número de documento de identidad	08467451
Datos de investigación	
Línea de investigación	No aplica
Grupo de investigación	No aplica
Agencia de financiamiento	Propio
Ubicación geográfica de la investigación	País: Perú Departamento: Lima Provincia: Lima Distrito: Cercado de Lima

	Jr. Carlos Amezaga No. 375 Universidad Nacional Mayor de San Marcos Latitud: -12.0564232 Longitud: -77.0843327
Año o rango de años en que se realizó la investigación	2021
URL de disciplinas OCDE	2.02.04 -- Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
Escuela Profesional de Ingeniería de Sistemas

Acta Virtual de Sustentación
del Trabajo de Suficiencia Profesional

Siendo las 15:15 horas del día 21 de diciembre del año 2021, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Lic. Romero Naupari Pablo Jesús (Presidente), Ing. Yáñez Durán Carlos Enrique (Miembro) y la Dra. Pró Concepción Luzmila Elisa (Miembro Asesor), usando la plataforma Meet (<https://meet.google.com/jjy-yahj-fza>), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: **“MIGRACIÓN DEL BACKEND DEL CANAL AGENTE DE UNA ENTIDAD FINANCIERA HACIA UNA ARQUITECTURA DE MICROSERVICIOS”**, por el Bachiller **Chipana Ramos Erick Herminio**; para obtener el Título Profesional de Ingeniero de Sistemas.

Acto seguido de la exposición del Trabajo de Suficiencia Profesional, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecidas por los miembros del Jurado.

El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el Bachiller obtuvo la nota de **17 DIECISIETE**.

A continuación el Presidente de Jurados el Lic. Romero Naupari Pablo Jesús, declara al Bachiller **Ingeniero de Sistemas**.

Siendo las 16:00 horas, se levantó la sesión.

Presidente

Lic. Romero Naupari Pablo Jesús

Miembro

Ing. Yáñez Durán Carlos Enrique

Miembro Asesor

Dra. Pró Concepción Luzmila Elisa

DEDICATORIA

A mis padres, por su apoyo incondicional desde que ingresé a la universidad hasta el día de hoy, sin ellos nada de esto hubiera sido posible.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

**Migración del Backend del Canal Agente de una Entidad Financiera hacia
una Arquitectura de Microservicios**

Autor : Erick Chipana Ramos

Asesor : Luzmila Elisa Pró Concepción

**Título : Trabajo de Suficiencia Profesional para optar el Título Profesional de
Ingeniero de Sistemas**

Fecha : Octubre del 2021

RESUMEN

El presente trabajo de experiencia profesional describe el rol y las tareas que ha desarrollado el autor durante el último año en el Squad Agentes de una entidad financiera. El proceso en sí consiste en la migración del backend de la aplicación del agente hacia una arquitectura de microservicios. La presente migración del backend de la aplicación agente se realizó debido a que la entidad financiera tiene como objetivo la renovación de todos los terminales POS usados actualmente en los agentes. Esta migración se hizo siguiendo el enfoque de microservicios, el cual fue propuesto por el equipo de arquitectura de la entidad financiera con el fin de lograr una alta disponibilidad, escalabilidad y mejorar la mantenibilidad de la aplicación. A nivel de negocio lo que se busca es mejorar la experiencia del usuario y generar afiliaciones de nuevos agentes.

Palabras clave: terminales POS, microservicios, backend, squad, canal agente.

MAJOR NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF SYSTEMS ENGINEERING
PROFESSIONAL SCHOOL OF SYSTEMS ENGINEERING

**Migration of the Backend of the Agent Channel of a Financial Institution
towards a Microservices Architecture**

Author : Erick Chipana Ramos

Adviser : Luzmila Elisa Pró Concepción

**Title : Professional Sufficiency Work for opt for the Professional Title of
Systems Engineer**

Fecha : October 2021

ABSTRACT

This work of professional experience describes the role and tasks that the author has developed during the last year in the Squad Agents of a financial institution. The process itself consists of migrating the backend of the agent application towards a microservices architecture. The present migration of the backend of the agent application was carried out because the financial institution aims to renew all the POS terminals currently used in the agents. This migration was done following the microservices approach, which was proposed by the architecture team of the financial institution in order to achieve high availability, scalability and improve the maintainability of the application. At the business level, what is sought is to improve the user experience and generate affiliations of new agents.

Keywords: POS terminals, microservices, backend, squad, agent channel.

ÍNDICE GENERAL

FICHA CATALOGRÁFICA.....	III
DEDICATORIA.....	IV
RESUMEN.....	V
ABSTRACT.....	VI
ÍNDICE GENERAL.....	VII
ÍNDICE DE TABLAS.....	IX
ÍNDICE DE FIGURAS.....	X
INTRODUCCIÓN.....	1
CAPÍTULO I - TRAYECTORIA PROFESIONAL.....	3
1.1 Presentación Profesional	3
1.2 Experiencia Profesional	3
1.3 Formación Académica	5
CAPÍTULO II - CONTEXTO EN QUE SE DESARROLLÓ LA EXPERIENCIA.....	6
2.1 Empresa - Actividad que realiza	6
2.2 Visión	6
2.3 Misión	7
2.4 Organización de la empresa	7
2.5 Área, cargo y funciones desempeñadas	8
2.6 Experiencia profesional realizada en la organización	9
CAPÍTULO III – ACTIVIDADES DESARROLLADAS.....	10
3.1 Situación Problemática	10
3.1.1 Definición del problema	11
3.2 Solución	12
3.2.1 Objetivos	12
3.2.2 Alcance	13
3.2.3 Etapas y metodologías	13
3.2.3.1 Etapas del proceso de migración	13
3.2.3.2 Marco de trabajo SCRUM	14
3.2.3.3 Herramientas de apoyo y lenguajes de programación	16
3.2.4 Fundamentos Utilizados	18
3.2.4.1 Terminales POS usados en el canal agentes	18
3.2.4.2 Diagrama de Arquitectura AS IS	19
3.2.4.3 Diagrama TO BE	20
3.2.4.4 Proyecto Nueva Plataforma Backend Agentes	21

3.2.4.5	Proyecto Nueva Plataforma Frontend Agentes	22
3.2.4.6	Diagrama Simplificado de Integración de los Proyectos Nueva Plataforma Backend y Frontend	24
3.2.4.7	Fundamentos Conceptuales	25
3.2.5	Implementación de las áreas, procesos, sistemas y sus buenas prácticas ..	26
3.2.5.1	Creación del contrato del API	26
3.2.5.2	Desarrollo y Pruebas del API	28
3.2.5.3	Pase a producción	30
3.3	Evaluación	34
CAPÍTULO IV – REFLEXIÓN CRÍTICA DE LA EXPERIENCIA		35
CAPÍTULO v – CONCLUSIONES Y RECOMENDACIONES		36
5.1	Conclusiones	36
5.2	Recomendaciones	37
5.3	Fuentes de Información	38
GLOSARIO		39
ANEXOS		41
ANEXO 1. Arquitectura AS IS Detallada		41
ANEXO 2. Microservicios desarrollados		42
ANEXO 3. Historias de usuario de un Sprint		43
ANEXO 4. Renovación de los terminales POS		44
ANEXO 5. Afiliaciones de nuevos agentes		44

ÍNDICE DE TABLAS

Tabla 1 : Experiencia Profesional	3
Tabla 2 : Formación Académica	5
Tabla 3 : Terminales POS usados a lo largo de los años.	18

ÍNDICE DE FIGURAS

Figura 1 : Organigrama de la Gerencia General Adjunta de Personas y Pequeños Negocios.	7
Figura 2 : Organigrama de la Tribu de Canales Alternativos.	8
Figura 3 : Agentes en un sector de Lima Metropolitana.	11
Figura 4 : Marco de trabajo Scrum	14
Figura 5 : Herramienta Utilizadas	17
Figura 6 : Diagrama de Arquitectura AS IS del canal Agente	19
Figura 7 : Diagrama de Arquitectura TO BE del canal Agente	20
Figura 8 : APK Agentes instalado en un Terminal POS NEW9220	22
Figura 9 : Polaris Cloud	23
Figura 10 : Polaris Core.....	23
Figura 11 : Integración Backend - Frontend	24
Figura 12 : Herramienta SwaggerHub	27
Figura 13 : Contrato Swaggerhub del API de Depósito.....	27
Figura 14 : Interfaz para la creación del proyecto Base de la API	28
Figura 15 : Codificación del API utilizando IntelliJ IDEA.....	29
Figura 16 : Despliegue del API en el ambiente de desarrollo	29
Figura 17 : Pruebas a nivel de POSTMAN.....	30
Figura 18 : Ticket Jira para el pase a producción.	31
Figura 19 : Flujo de un Ticket Jira.....	33
Figura 20 : Arquitectura AS IS detallado	41
Figura 21 :Microservicios desplegados en la suscripción Azure.....	42
Figura 22 : Tablero Kanban.....	43
Figura 24 : Renovación del parque de agentes	44
Figura 25 : Diagrama de barras de afiliaciones de agentes	44

INTRODUCCIÓN

El presente trabajo de suficiencia profesional describe el proceso de migración hacia una arquitectura de microservicios del backend de la aplicación del canal agentes de la entidad financiera. Esta migración tiene como finalidad la búsqueda de la escalabilidad de la aplicación, la interoperabilidad con otras aplicaciones, la disminución del tiempo de respuesta de las transacciones y mejorar la disponibilidad de la aplicación del canal . Toda esta migración va de la mano con la renovación de los terminales POS usados por los negocios que cumplen el rol de agentes.

Toda esta renovación de los terminales POS se debe a que los antiguos presentan averías cada cierto tiempo, por otro lado se hace complicado el mantenimiento de estos equipos ya que cuentan con un sistema operativo antiguo que no tiene soporte en la actualidad, por último en estos equipos no se pueden incluir nuevas funcionalidades lo cual representa un problema para el cumplimiento de los objetivos de la entidad financiera.

Este trabajo de suficiencia representa el proyecto en el que participó el autor de manera activa durante el último año y a la cual considera la más importante. El presente trabajo de suficiencia está organizado de la siguiente manera:

En el CAPITULO I se detalla de manera cronológica la trayectoria profesional del autor, funciones y las principales actividades desarrolladas.

En el CAPITULO II se describe una breve historia de la entidad financiera donde el autor realizó el proyecto, además se describe su estructura orgánica, la visión, la misión, las funciones desempeñadas y la experiencia profesional del autor.

En el CAPITULO III se detalla el trabajo realizado, el cual se refiere al proyecto de migración del backend del canal agentes hacia una arquitectura de microservicios, se describe la metodología utilizada, se especifican los fundamentos , el proceso de desarrollo y se resaltan los puntos más importantes que se presentaron durante la realización de este trabajo.

En el CAPITULO IV se describe el aporte del autor de este trabajo, la experiencia obtenida, los conocimientos que demandó, así como el desarrollo profesional que obtuvo con la realización del presente trabajo.

En el CAPITULO V se describen las conclusiones y recomendaciones resultantes del proyecto.

CAPÍTULO I - TRAYECTORIA PROFESIONAL

1.1 Presentación Profesional

El autor del presente trabajo es Bachiller en Ingeniería de Sistemas, actualmente en el rol de Analista TI para la empresa Tata Consultancy Services. Cuenta con una experiencia laboral de más de 7 años en desarrollo de software como programador backend y frontend. Tiene experiencia laboral en el sector público y privado, ha participado en proyectos para América Móvil SAC, entidades financieras como el Banco de Crédito del Perú y el Banco de la Nación , además ha realizado consultorías para el Ministerio de Justicia y Derechos de Humanos.

1.2 Experiencia Profesional

Tabla 1 : Experiencia Profesional

Tata Consultancy Services	
Agosto 2020 – Actualidad	
Cargo	Analista TI
Cliente	BCP
Funciones	Desarrollar microservicios utilizando el framework personalizado de la entidad financiera. Migración hacia las nuevas versiones del framework de la entidad financiera. Desarrollar y publicar los contratos de los microservicios utilizando la herramienta SwaggerHub. Desarrollar microservicios para la interconexión con la entidad financiera Mi Banco. Realizar y actualizar toda la documentación.
Agilsoft SAC	
Agosto 2018 – Julio 2020	
Cargo	Programador Fullstack
Cliente	Novatronic
Funciones	Desarrollar el Sistema de Afiliación del Banco de la Nación. Desarrollar el Sistema de Valorados del Banco de la Nación.

	<p>Desarrollar el Sistema de Planificación de Recursos Empresariales de la empresa OILBANK.</p> <p>Desarrollar Servicios Web SOAP para el Banco de la Nación.</p> <p>Desarrollar Servicios Web SOAP para América Móvil.</p> <p>Realizar y actualizar toda la documentación.</p>
Ministerio de Justicia y Derechos Humanos	
Febrero 2018 – Julio 2018	
Cargo	Desarrollador GIS
Cliente	Ministerio de Justicia y Derechos Humanos
Funciones	<p>Desarrollar el lado backend y frontend del proyecto denominado "Mapa de la Justicia".</p> <p>Consumo de Servicios REST de geoprocesamiento alojados en un ArcGIS Server.</p> <p>Desarrollar el frontend del "Mapa de la Justicia" utilizando Angular y Dojo.</p> <p>Realizar y actualizar toda la documentación.</p>
Agilsoft SAC	
Febrero 2017 – Diciembre 2017	
Cargo	Programador FullStack
Cliente	Novatronic
Funciones	<p>Migrar del sistema SIXs (SIX-CA y SIX -TCL) de la empresa Novatronic.</p> <p>Desarrollar el sistema de envíos de mensaje de texto para Interbank.</p> <p>Desarrollar y mantener el sistema Home Banking de Caja Piura.</p> <p>Realizar y actualizar toda la documentación de los sistemas desarrollados.</p>
Clínica Vesalio	
Diciembre 2014 – Diciembre 2016	
Cargo	Programador .NET/C#
Cliente	Clínica Vesalio
Funciones	<p>Realizar el análisis y levantamiento de requerimientos.</p> <p>Desarrollar y mantener de los distintos sistemas de la clínica tales como: Sistema Hospitalario, Sistema Ambulatorio, Sistema de Pagos de Planilla a Médicos, Sistema de Quejas y Reclamos, Sistema de Farmacia, Sistema de Exámenes y Procedimientos, Sistema del Centro Médico, etc.</p> <p>Realizar y actualizar toda la documentación asociada a los sistemas de la clínica.</p>

Nota: Elaboración propia.

1.3 Formación Académica

Tabla 2 : Formación Académica

2009 - 2016	Universidad Nacional Mayor de San Marcos - Facultad de Ingeniería de Sistemas e Informática Grado académico de Bachiller en Ingeniería de Sistemas
2013	Sistemas UNI Curso PHP Nivel I - Programación Curso PHP Nivel I - DAO
2017	Cibertec Curso Java Advanced Developer 8.0
2018	CJAVA Curso Java Programmer 8.0
2018	Galaxy Training Curso Servicios Rest en Java
2019	Mitocode Curso Java/Angular Full Stack Developer

Nota: Elaboración propia.

CAPÍTULO II - CONTEXTO EN QUE SE DESARROLLÓ LA EXPERIENCIA

2.1 Empresa - Actividad que realiza

La entidad financiera es una institución peruana y es el proveedor líder de servicios financieros en el país, representando el principal activo del grupo financiero Credicorp, el holding financiero más importante del Perú. El objeto social de la entidad financiera es el de favorecer el desarrollo de las actividades comerciales y productivas del país, que incluyen la actividad bancaria comercial y de ahorros.

A través de sus divisiones de Banca Corporativa y Banca Empresa proveen servicios especialmente diseñados para clientes corporativos y empresas medianas, mientras que desde Banca Minorista atienden a pequeñas empresas y clientes individuales. Las actividades de su sector se rigen por la Ley 267025, norma que tiene como objetivo promover al funcionamiento de un sistema financiero y un sistema de seguros competitivos, sólidos y confiables, que contribuyan al desarrollo nacional.

2.2 Visión

“Ser la empresa peruana que brinda la mejor experiencia a los clientes. Simple, cercana y oportuna”.

“Ser la comunidad laboral de preferencia en el Perú, que inspira, potencia y dinamiza a los mejores profesionales”.

“Ser referentes regionales en gestión empresarial potenciando nuestro liderazgo histórico y transformador de la industria financiera en el Perú”. (Financiera, 2021).

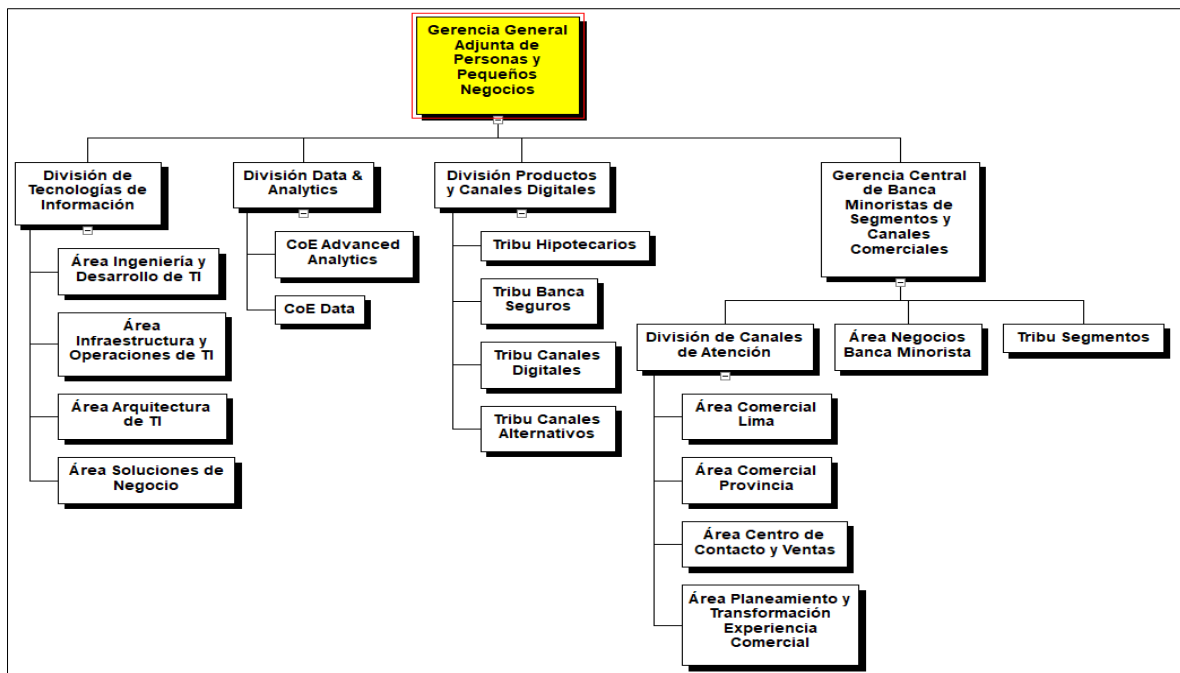
2.3 Misión

“Estar siempre contigo, alentando y transformando tus sueños y planes en realidad y con el Perú, construyendo su historia de desarrollo y superación”.
(Financiera, 2021).

2.4 Organización de la empresa

A continuación, se muestra solo una parte del organigrama de la entidad financiera la cual es el área de Gerencia General Adjunta de Personas y Pequeños Negocios a la que el autor considera más importante para el presente trabajo de suficiencia .

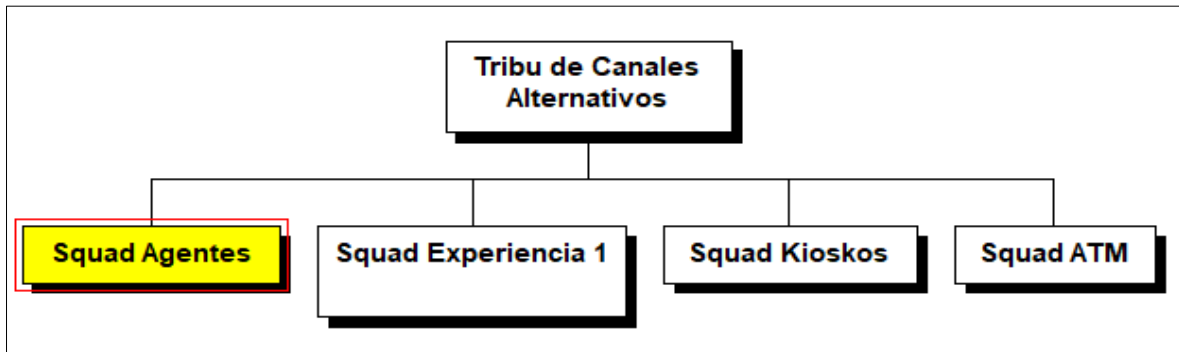
Figura 1 : Organigrama de la Gerencia General Adjunta de Personas y Pequeños Negocios.



Nota: Elaboración propia.

La experiencia profesional se desarrolló en el Squad Agentes de la Tribu de Canales Alternativos, el organigrama se muestra a continuación:

Figura 2 : Organigrama de la Tribu de Canales Alternativos.



Nota: Elaboración propia.

2.5 Área, cargo y funciones desempeñadas

Área: Tribu Canales Alternativos, Squad Agentes.

Cargo: Programador Backend.

Funciones Desempeñadas:

- Desarrollar microservicios siguiendo los lineamientos de la entidad financiera.
- Migrar los microservicios que componen el backend de la aplicación del canal agente hacia la última versión del framework de la entidad financiera.
- Mantener los microservicios corrigiendo bugs o agregando nuevas funcionalidades.
- Desarrollar y publicar los contratos de los microservicios.
- Actualización de la documentación de las aplicaciones del canal Agentes.
- Realizar pruebas de stress a los diferentes microservicios en los ambientes de desarrollo y certificación.
- Realizar y gestionar los pases a producción de los nuevos microservicios.
- Brindar soporte 24x7 al canal agentes.

2.6 Experiencia profesional realizada en la organización

El autor como team member del Squad Agentes tuvo las siguientes experiencias:

- Participó en capacitaciones relacionadas al marco de trabajo ágil SCRUM esto con el fin de entender la forma de trabajo de un equipo ágil en la entidad financiera.
- Realizó el desarrollo y migración de los microservicios utilizando el framework de la entidad financiera.
- Realizó la migración del backend de las aplicaciones del canal agentes.
- Participó en el proyecto de interconexión con otras entidades financieras logrando así poder realizar Depósitos y Retiros con la entidad financiera Mi Banco.
- Participó en la migración hacia el nuevo API Management del canal Agentes.
- Participó en las distintas reuniones realizadas por la arquitecta del canal, así como con el equipo de arquitectura cloud.

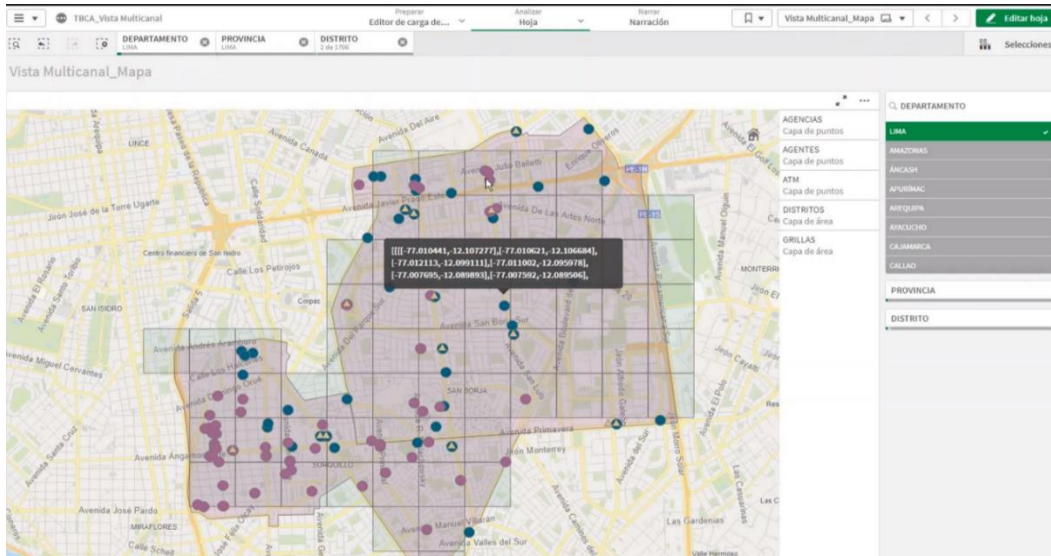
CAPÍTULO III – ACTIVIDADES DESARROLLADAS

3.1 Situación Problemática

El canal de agentes actualmente cuenta con muchas operaciones disponibles en sus agentes tales como depósito, retiro, pago de servicios, consulta de saldos y movimientos, emisión y descargo de giros, transferencias, etc. La entidad financiera tiene planeado agregar nuevas funcionalidades a los terminales POS usados por los agentes, esto con el fin de sumar a más negocios como agentes, marcar una diferenciación con los agentes de otras entidades financieras y lograr un mayor ingreso económico.

Para cumplir con este objetivo la entidad financiera necesita contar con una arquitectura adecuada lo cual le garantice una alta disponibilidad para lograr la satisfacción de los agentes y por otro lado satisfacer la necesidad de los usuarios finales. Todo este proceso de migración del backend se hará conjuntamente con la migración del frontend y con la renovación de los equipos terminales POS. Toda esto conlleva un severo cambio por lo que se está realizando de manera gradual, esto con el fin de no afectar las transacciones diarias de los agentes.

Figura 3 : Agentes en un sector de Lima Metropolitana.



Nota: Tomada de la Entidad Financiera.

3.1.1 Definición del problema

Por lo explicado en el punto anterior, se puede definir el problema principal de la siguiente manera:

Terminales POS antiguos que no permiten la inclusión de nuevas funcionalidades, el mantenimiento resulta complejo y no brindan una buena experiencia a los agentes de la entidad financiera.

3.2 Solución

El presente trabajo de suficiencia busca resolver el problema mencionado en el punto anterior mediante la migración del backend del aplicativo del canal agentes de una entidad financiera hacia una arquitectura de microservicios.

3.2.1 Objetivos

Objetivo General

Realizar la migración del backend de la aplicación del canal agentes de una entidad financiera hacia una arquitectura de microservicios.

Objetivo Específicos

- Crear el contrato swagger del API utilizando la herramienta SwaggerHub.
- Crear el código fuente de la API utilizando el framework de la entidad financiera y siguiendo los lineamientos de desarrollo.
- Consumir las últimas versiones de los servicios de la entidad financiera tales como Cards, Accounts, Persons, etc.
- Realizar pruebas de la transacción migrada en el ambiente de desarrollo en conjunto con el analista de calidad.
- Realizar la actualización de la documentación, diagramas, etc.
- Realizar pruebas de la transacción migrada en el ambiente de certificación en conjunto con el analista de calidad.
- Gestionar el pase a producción de la transacción migrada.

3.2.2 Alcance

Alcance Funcional

El alcance del presente trabajo de suficiencia profesional es realizar la migración del backend de la aplicación del canal agentes y su posterior despliegue en la nube de Azure.

3.2.3 Etapas y metodologías

3.2.3.1 Etapas del proceso de migración

En cuanto a las etapas de desarrollo de este proceso de migración podemos resumir las siguientes:

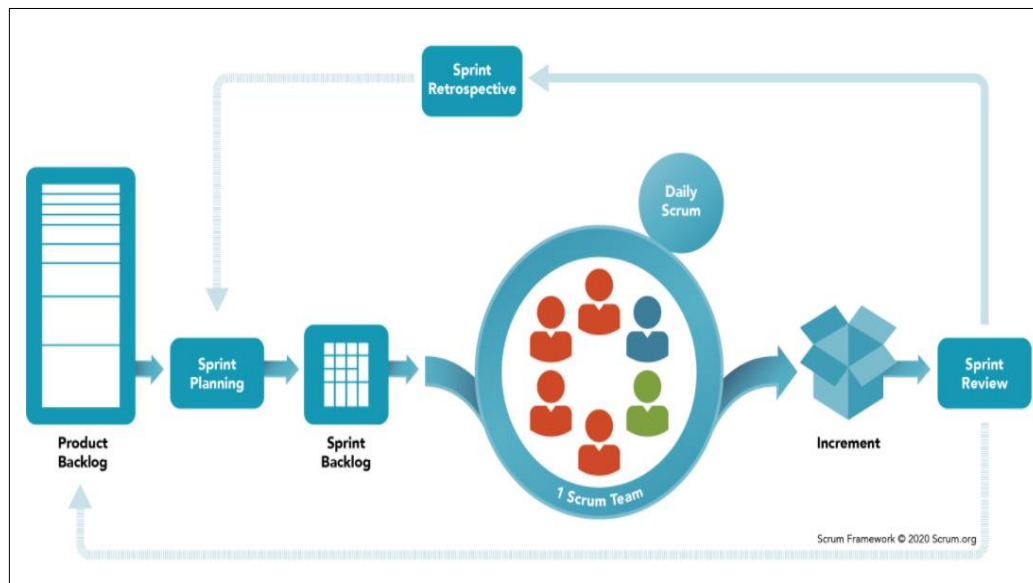
1. Creación, aprobación y publicación de los contratos de las API's con la herramienta SwaggerHub.
2. Desarrollo del proyecto base de microservicio utilizando el framework de desarrollo de la entidad financiera.
3. Análisis y recolección de data para las pruebas del microservicio en el ambiente local.
4. Despliegue y pruebas en el ambiente de desarrollo.
5. Presentación y cumplimiento de los lineamientos de seguridad de la entidad financiera.
6. Actualización de toda la documentación asociada.
7. Despliegue y pruebas en el ambiente de certificación.
8. Gestión y ejecución del pase a producción.

3.2.3.2 Marco de trabajo SCRUM

Las etapas detalladas en el punto anterior fueron desarrolladas siguiendo la metodología SCRUM. Schwaber y Sutherland (2016) nos dan la siguiente definición :

“Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas”.

Figura 4 : Marco de trabajo Scrum



Nota : Tomada de (Scrum.org, 2021)

El uso de este marco de trabajo fue usado para este proyecto ya que la entidad financiera exigía rapidez en los resultados, pero a su vez flexibilidad ya que los requerimientos iban cambiando en el tiempo de acuerdo al análisis realizado por los especialistas del negocio en conjunto con el product owner.

El proyecto tuvo una duración de 24 sprints los cuales tuvieron una duración de 2 semanas cada uno y fueron divididos en 4 quarters o trimestres. Entre los principales roles de este proyecto podemos describir los siguientes:

- **Product Owner:** “Es el responsable de maximizar el valor del producto y el trabajo del Equipo de Desarrollo. El cómo se lleva a cabo esto podría variar ampliamente entre distintas organizaciones, Equipos Scrum e individuos”. (Schwaber & Sutherland, 2016).
- **Agile Coach:** Es la persona responsable de liderar la adopción del método ágil en la organización, guiando a sus miembros a través del proceso de implantación, poniendo a disposición de la organización entrenamiento, herramientas y conocimientos necesarios para que se pueda acrecentar todo el potencial de la agilidad.
- **Team member:** “Son los profesionales que realizan el trabajo de entregar un incremento de producto ‘Terminado’ que potencialmente se pueda poner en producción al final de cada Sprint. Solo los miembros del Equipo de Desarrollo participan en la creación del incremento”. (Schwaber & Sutherland, 2016).

Entre los beneficios del uso de Scrum reflejados en este proyecto tenemos:

- **Desarrollo ágil:** Scrum propició el desarrollo rápido y flexible, esto aseguró la entrega de un producto muy alineado a las necesidades y deseos del cliente, además se optimizó el tiempo, los recursos y los esfuerzos durante la etapa de desarrollo.
- **Alta calidad de software:** Scrum propició el desarrollo de software de alta calidad ya que este marco de trabajo se enfoca en las necesidades del cliente, en la priorización de los módulos que verdaderamente entregan valor. Adicionalmente el equipo de desarrollo contó con constante

retroalimentación sobre las funcionalidades esperadas y los cambios que podrían darse en el camino.

- **Equipos integrados:** Scrum propició el trabajo en de manera conjunta y unida en el squad, esto debido a que los procesos de este marco propician esta forma de trabajo, un claro ejemplo fueron las reuniones que se llevaron a cabo de manera diaria, la constante retroalimentación entre los integrantes del squad, el claro conocimiento de las metas, avances y prioridades del proyecto, etc.
- **Fácil escalabilidad:** Los procesos se manejaron de manera iterativa y en periodos de tiempo específicos, esto hizo que el equipo de desarrollo se centre en funcionalidades puntuales por cada periodo. Los beneficios se vieron reflejados en la entrega de productos que dieron valor al cliente y además permitió al equipo la flexibilidad de poder realizar cambios a las funcionalidades de acuerdo a las necesidades cambiantes.

3.2.3.3 Herramientas de apoyo y lenguajes de programación

Entre las herramientas utilizadas para este proyecto tenemos siguientes:

- **Confluence:** es el software que actualmente usa la entidad financiera para compartir toda la información con relación a los procesos de desarrollo, lineamientos, herramientas de tecnologías de información usadas, etc.
- **IntelliJ IDEA:** entorno de desarrollo integrado que se usó para el desarrollo de las API's.
- **Jira:** es el software que se usó para la gestión de cada sprint del proyecto, en esta herramienta se registraron las historias de usuario de cada sprint, así como los responsables de cada una.
- **GitKraken:** es una herramienta para el control de versiones la cual posee una interfaz amigable la cual facilitó muchas de las tareas de versionamiento.

- **Jenkins:** es la herramienta de integración continua que se usó para el despliegue de los microservicios en los distintos ambientes.
- **Bitbucket:** es la herramienta que se usó para la gestión y control del código fuente de las distintas API's desarrolladas.
- **Postman:** es una aplicación que fue usada como cliente REST para la realización de las pruebas a las API's desarrolladas.

El lenguaje de programación utilizado en el proyecto fue Java 11.

Figura 5 : Herramienta Utilizadas



Nota: Elaboración propia.

3.2.4 Fundamentos Utilizados

En primer lugar, daremos un breve detalle de algunos conceptos y fundamentos necesarios para entender el proyecto. Posteriormente explicaremos los proyectos que existen actualmente en el canal.

3.2.4.1 Terminales POS usados en el canal agentes

El proveedor de los terminales POS de la entidad financiera es DIEBOLD el cual además se encarga del mantenimiento de estos terminales. A lo largo de los años hasta la actualidad se han usado 4 tipos de terminales, siendo el modelo de terminal NEW9220 el de última generación y al que se busca migrar en un corto plazo.

Tabla 3 : Terminales POS usados a lo largo de los años.

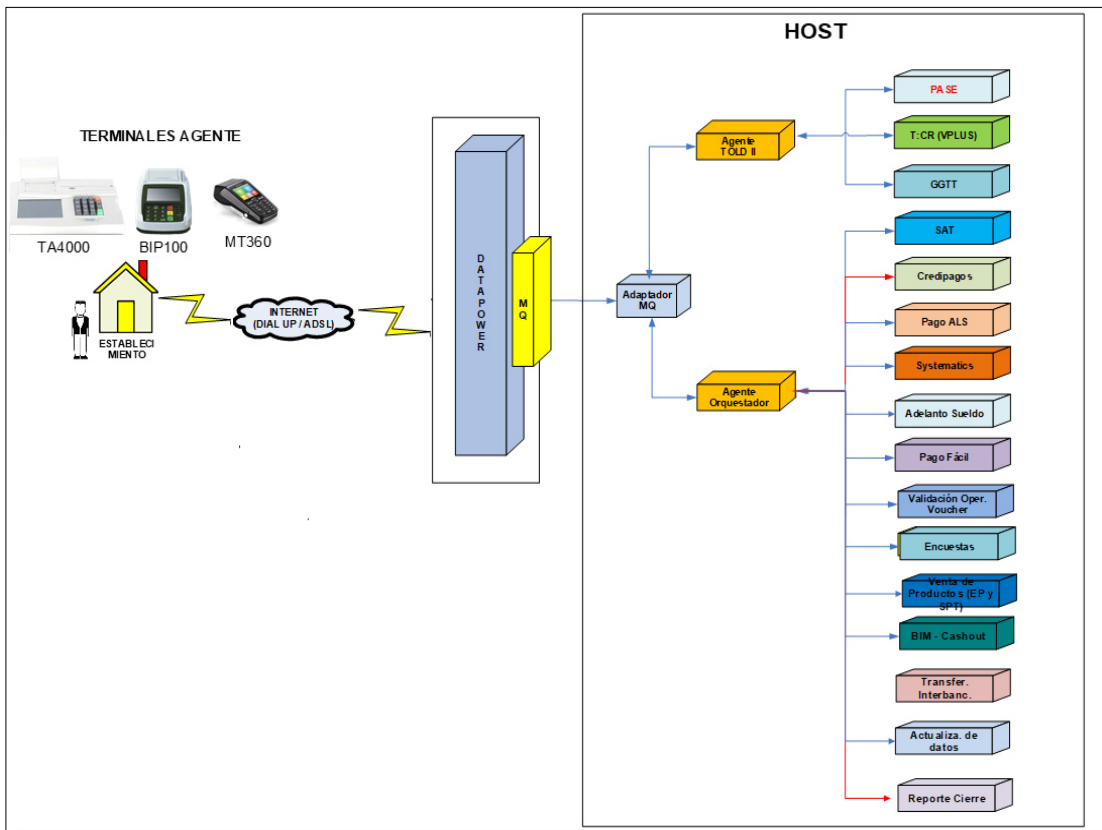
Hardware	Sistema Operativo	Base de datos	Dispositivo
TA4000	Windows CE 4.2	SQL CE	 TA4000
BIP100	Windows CE 5.0	SQL CE	 BIP100
MT360	Windows CE 5.0	SQL CE	 MT360
NEW9220	Android 5.0	SQL CE	 NEW9220

Nota: Elaboración propia.

3.2.4.2 Diagrama de Arquitectura AS IS

Esta arquitectura corresponde al antiguo Agente, es decir es decir a los terminales MT360 y los anteriores a éstos, toda esta arquitectura fue diseñada por especialistas en Mainframe de la entidad financiera y diferentes proveedores en conjunto, está compuesta por diferentes componentes legados y por programas los cuales residen en el Mainframe.

Figura 6 : Diagrama de Arquitectura AS IS del canal Agente



Nota: Tomada de la Entidad Financiera.

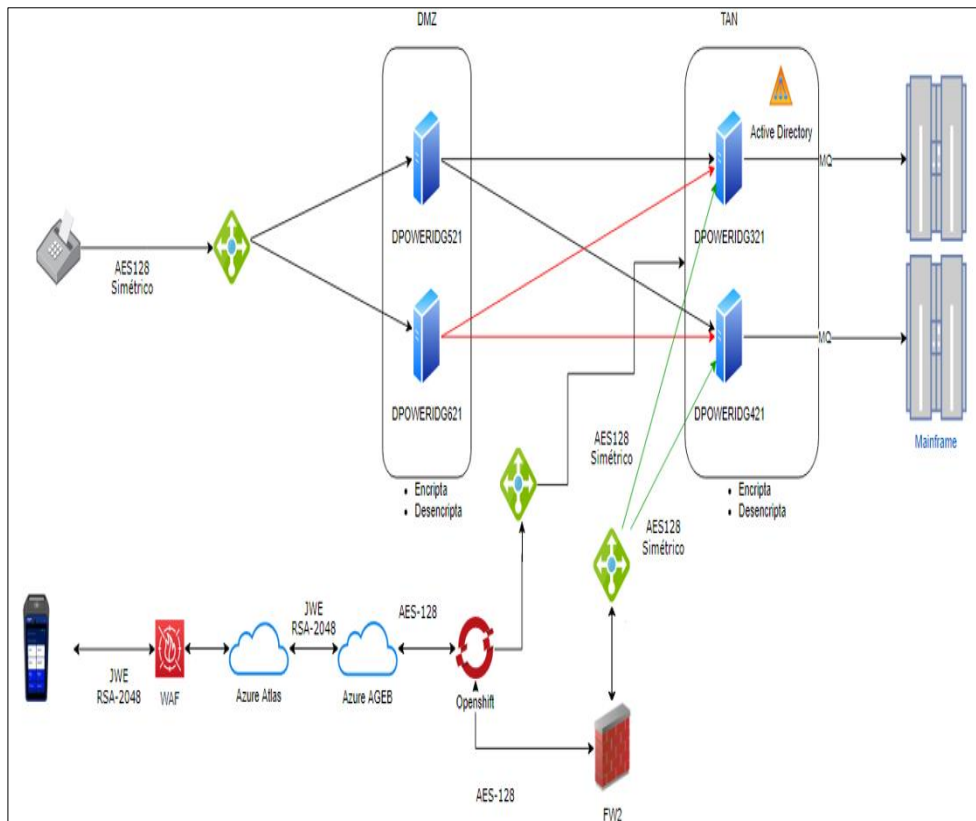
Como se puede observar en la figura anterior los terminales POS envían la información directamente al componente Datapower vía internet. Luego Datapower envía la información a un Middleware de mensajería (IBM MQ), la cual se comunica con un Adaptador MQ para

finalmente enviar la información al Mainframe para que la transacción sea ejecutada por el programa correspondiente. Para mayor información del diagrama de arquitectura, revisar Anexo 1.

3.2.4.3 Diagrama TO BE

Se realizó la migración del backend siguiendo el enfoque de microservicios y se realizó el despliegue de las API's en una suscripción de la nube de Azure correspondiente al canal.

Figura 7 : Diagrama de Arquitectura TO BE del canal Agente



Nota: Tomada de la Entidad Financiera.

En resumen, el flujo de la figura anterior es el siguiente:

1. El dispositivo POS envía la petición cifrada al backend, pero para esto primero debe pasar el firewall de aplicaciones web (WAF).
2. La petición es autenticada y autorizada por el API Gateway el cual se encuentra desplegado en una suscripción de Azure llamada Atlas.
3. Luego la petición llega a la suscripción de AGEB a la cual pertenece el canal Agentes y en el que se encuentran desplegados todos los microservicios que componen el backend.
4. Finalmente, la comunicación con el mainframe, que es quien ejecuta la transacción, se realiza mediante el componente DataPower.

3.2.4.4 Proyecto Nueva Plataforma Backend Agentes

Es el proyecto que va por detrás del APK Transaccional desarrollado por el proveedor WPOSS. Este proyecto nace para complementar al proyecto Nueva Plataforma de Agente Frontend, está compuesto por los distintos microservicios desarrollados con el Framework de la entidad financiera ya que su uso es un lineamiento de desarrollo.

Se crearon 15 API's en total aproximadamente, cada API corresponde a una funcionalidad del agente (APIS UX) o puede corresponder a un api de soporte (APIS BA) las cuales son componentes Cross a las demás API's.

Las API's en su gran mayoría fueron desplegadas en AKS y algunas otras en la plataforma OpenShift de la entidad financiera. Cada API tiene entre 4 y 8 endpoints o recursos, estos a su vez consumen otras API's de negocio de la entidad financiera tales como: Cards, Accounts, Persons, etc.

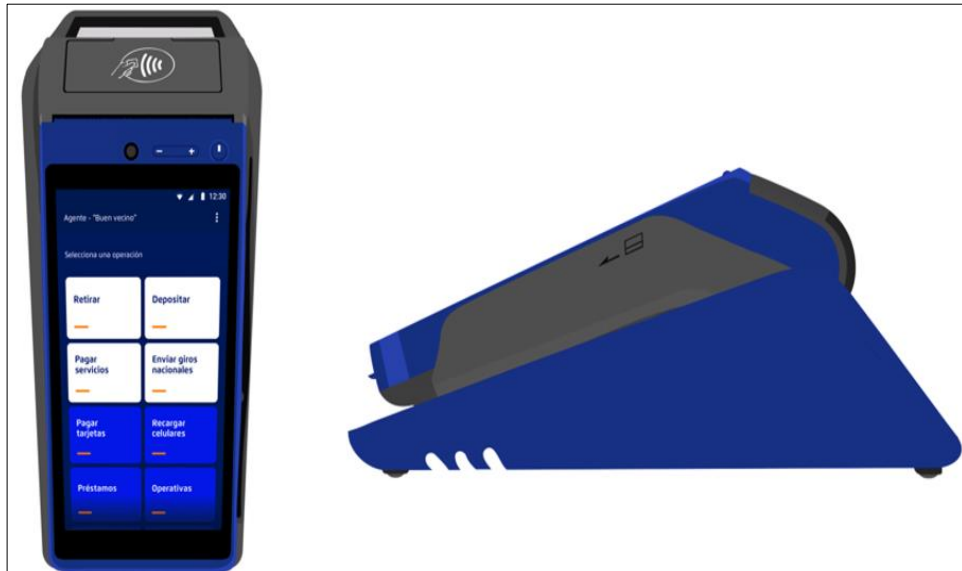
3.2.4.5 Proyecto Nueva Plataforma Frontend Agentes

Este proyecto nace debido a la renovación de los terminales POS. El nuevo aplicativo (APK) fue desarrollado por la empresa WPOSS como mencionamos anteriormente. Ambos proyectos (backend y frontend) se necesitan entre sí para poder llevar a cabo las funcionalidades del nuevo agente.

Este proyecto está compuesto por el APK Agentes de la nueva plataforma y por los módulos de administración de los terminales los cuales son los SAAS: Polaris Cloud y Polaris Core. A continuación, describiremos las partes involucradas en este proyecto:

APK Agentes: Esta aplicación viene ser el frontend de canal agentes, la cual es instalada en un dispositivo de Modelo NEW9220 con la versión Android 5.0 como sistema operativo.

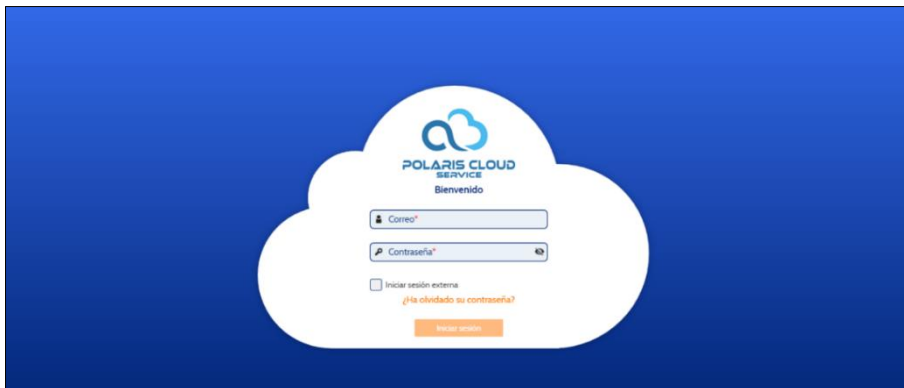
Figura 8 : APK Agentes instalado en un Terminal POS NEW9220



Nota: Tomada de la Entidad Financiera.

Polaris Cloud: En este módulo se ven temas relacionados como la habilitación de ambientes (Desarrollo, Certificación y Producción), integración con Azure Active Directory, carga del APK transaccional y sus respectivas versiones, renovación de certificados, etc.

Figura 9 : Polaris Cloud



Nota: Tomada de la Entidad Financiera.

Polaris Core: En este módulo se ven temas como la gestión comercial, gestión de incidencias, gestión de inventarios y configuraciones adicionales de los terminales POS.

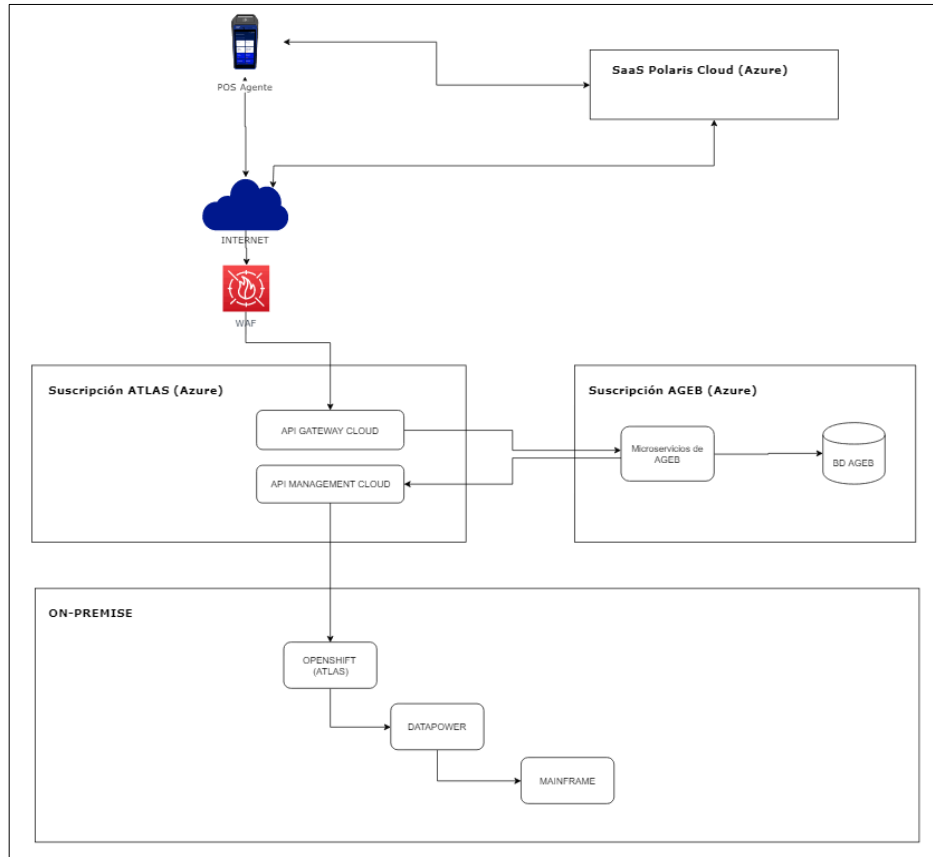
Figura 10 : Polaris Core



Nota: Tomada de la Entidad Financiera.

3.2.4.6 Diagrama Simplificado de Integración de los Proyectos Nueva Plataforma Backend y Frontend

Figura 11 : Integración Backend - Frontend



Nota: Elaboración propia.

En el gráfico anterior se puede detallar el siguiente flujo:

1. El frontend descarga las últimas configuraciones de Polaris Cloud.
2. El frontend se comunica en primer lugar con la suscripción Atlas pasando previamente por el firewall de aplicaciones web (WAF).
3. Dentro de la suscripción Atlas se encuentran el API Gateway el cual se encarga de autorizar y autenticar las distintas peticiones del frontend.

4. Una vez autorizado y autenticado la petición, ésta llega a la suscripción AGEB que es donde se encuentran desplegados los microservicios del canal agentes.
5. Finalmente, la comunicación con el mainframe se da mediante el componente Datapower.

3.2.4.7 Fundamentos Conceptuales

API

“Las interfaces de programación de aplicaciones exponen los servicios o los datos proporcionados por una aplicación de software a través de un conjunto de recursos predefinidos, como métodos, objetos o URIs “ (Stylos, Faulring, Yang, & Myers, 2009).

Framework

“Es un conjunto de clases de objetos que colaboran para llevar a cabo un conjunto de responsabilidades; reutiliza tanto el diseño de alto nivel de un programa como su implementación”. (Schmid, 1996).

Microservicio

“Es una arquitectura nativa de la nube que tiene como objetivo realizar sistemas de software como un paquete de pequeños servicios, cada uno de ellos implementado de forma independiente“. (Balalaie, Heydarnoori, & Jamshidi, 2016).

“Puede verse como una técnica para desarrollar aplicaciones de software que, heredando principios y conceptos del estilo Arquitectura Orientada a Servicios (SOA), permite estructurar una aplicación basada en servicios

como una colección de pequeños servicios de software débilmente acoplados”. (Dragoni, et al., 2017)

Cloud

“Es un término que se utiliza para describir una red global de servidores, cada uno con una función única. La nube no es una entidad física, sino una vasta red de servidores remotos en todo el mundo que están conectados y destinados a funcionar como un único ecosistema”. (Microsoft, 2021)

Backend

Es la capa que se encarga de procesar toda la información que será enviada al frontend, contiene la lógica del negocio y es el encargado del acceso a la base de datos.

3.2.5 Implementación de las áreas, procesos, sistemas y sus buenas prácticas

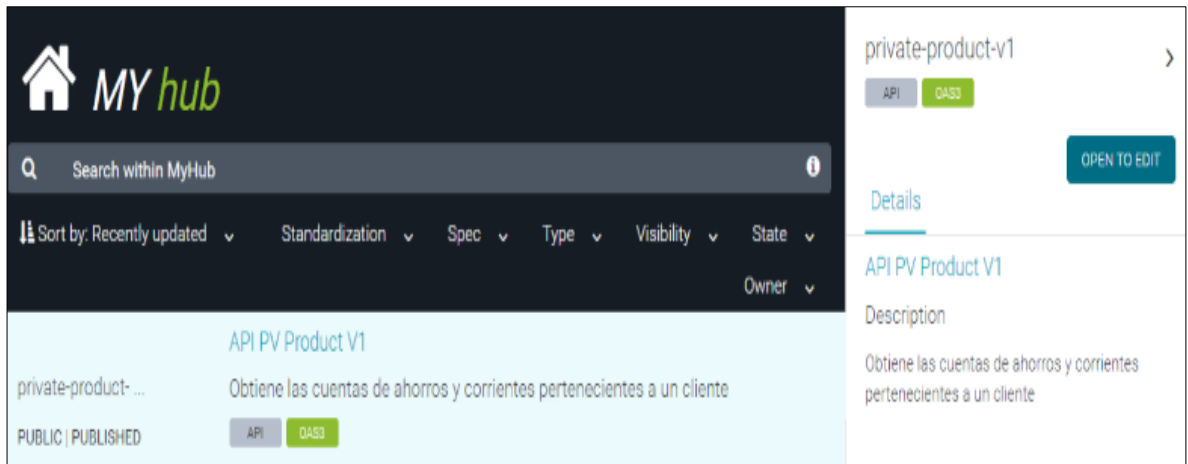
Para este punto detallaremos el proceso de desarrollo de una API.

3.2.5.1 Creación del contrato del API

Actualmente la entidad financiera cuenta con una iniciativa denominada Design First, mediante este modelo de construcción de API's, se debe primero realizar el diseño de la API antes de realizar el desarrollo del microservicio. El diseño de la API está representado por el contrato swagger, el cual nos brindará el detalle de los endpoints del API, definiciones de los modelos del API y otras especificaciones. A través del contrato swagger podremos conocer funcionalidades que expondrá el API antes de iniciar su desarrollo.

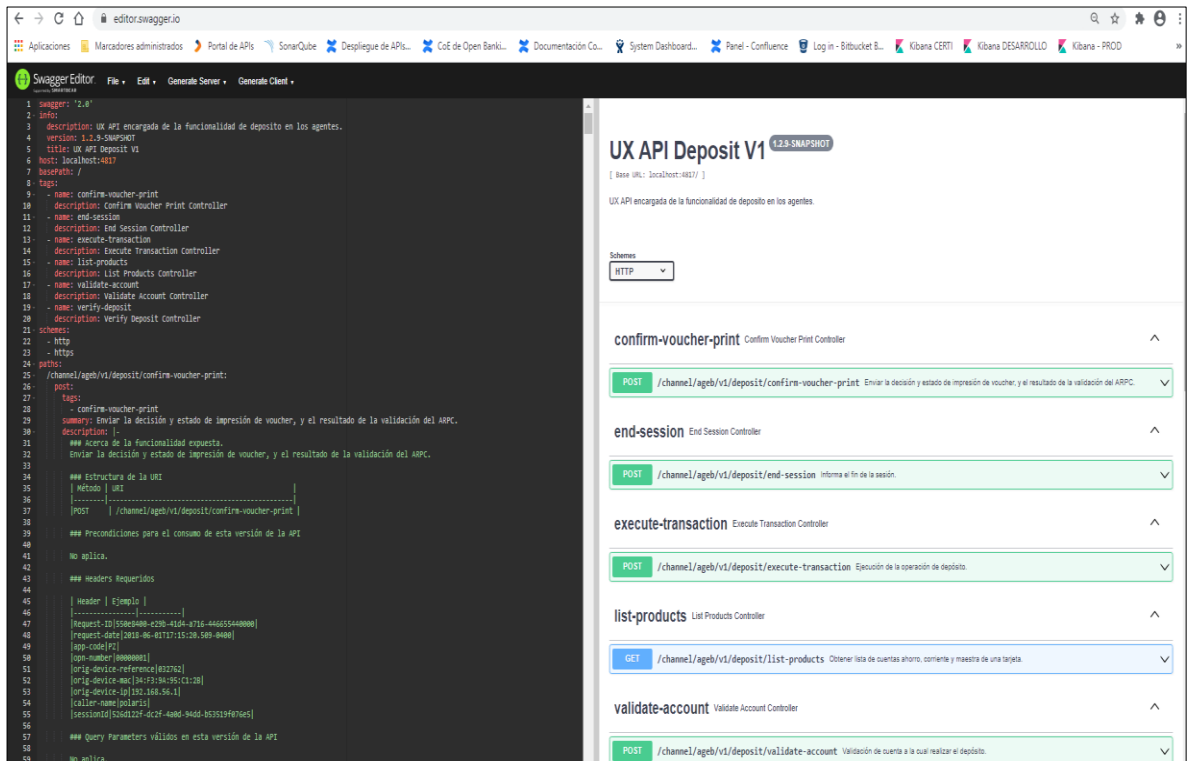
El contrato swagger es creado mediante la herramienta SwaggerHub, este proceso de creación es revisado de manera constante por el equipo denominado Gobierno de API's el cual una vez que no encuentre observaciones en el contrato swagger dará su visto bueno y se procederá a la publicación del contrato y posteriormente al desarrollo de la misma.

Figura 12 : Herramienta SwaggerHub



Nota: Tomada de la Entidad Financiera.

Figura 13 : Contrato Swaggerhub del API de Depósito.

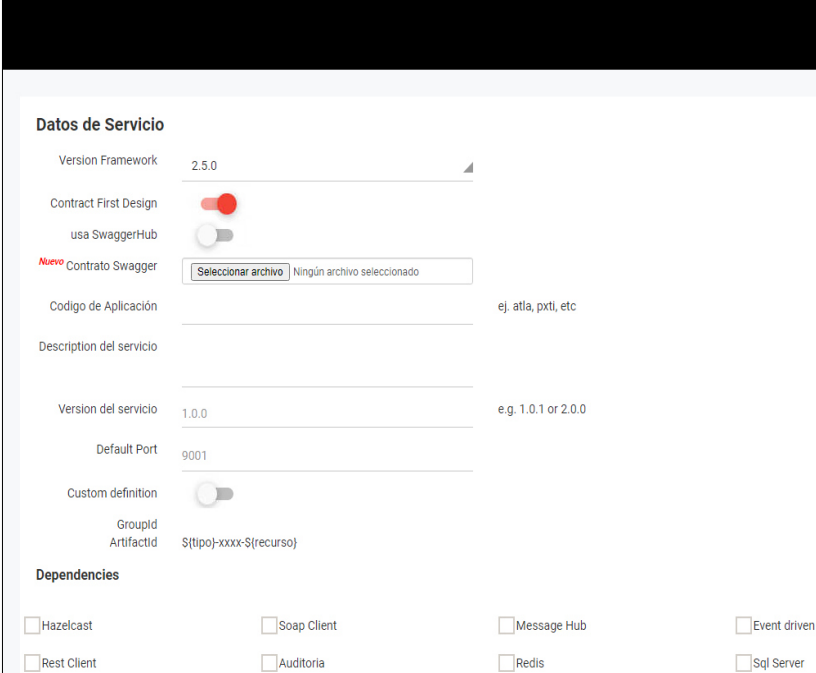


Nota: Tomada de la Entidad Financiera.

3.2.5.2 Desarrollo y Pruebas del API

Una vez finalizada la creación del contrato swagger se procede a la creación del proyecto base de la API el cual debe ser creado con el framework de la entidad financiera. Para la creación de este proyecto base se cuenta con una interfaz gráfica en la cual el desarrollador debe de realizar la importación del contrato swagger creado anteriormente, completar la información del API y finalmente seleccionar las dependencias necesarias.

Figura 14 : Interfaz para la creación del proyecto Base de la API



The screenshot displays a web-based form for creating an API project. It is divided into two main sections: 'Datos de Servicio' and 'Dependencies'.

Datos de Servicio:

- Version Framework:** A dropdown menu set to '2.5.0'.
- Contract First Design:** A toggle switch that is turned on (red).
- usa SwaggerHub:** A toggle switch that is turned off (grey).
- Nuevo Contrato Swagger:** A file selection button labeled 'Seleccionar archivo' with the text 'Ningún archivo seleccionado' next to it.
- Código de Aplicación:** A text input field with a placeholder example 'ej. atla, pxlt, etc'.
- Description del servicio:** A text input field.
- Version del servicio:** A text input field with a placeholder example 'e.g. 1.0.1 or 2.0.0'.
- Default Port:** A text input field set to '9001'.
- Custom definition:** A toggle switch that is turned off (grey).
- Groupid / Artifactid:** A text input field with a placeholder pattern 'S{tipo}-xxxx-S{recurso}'.

Dependencies:

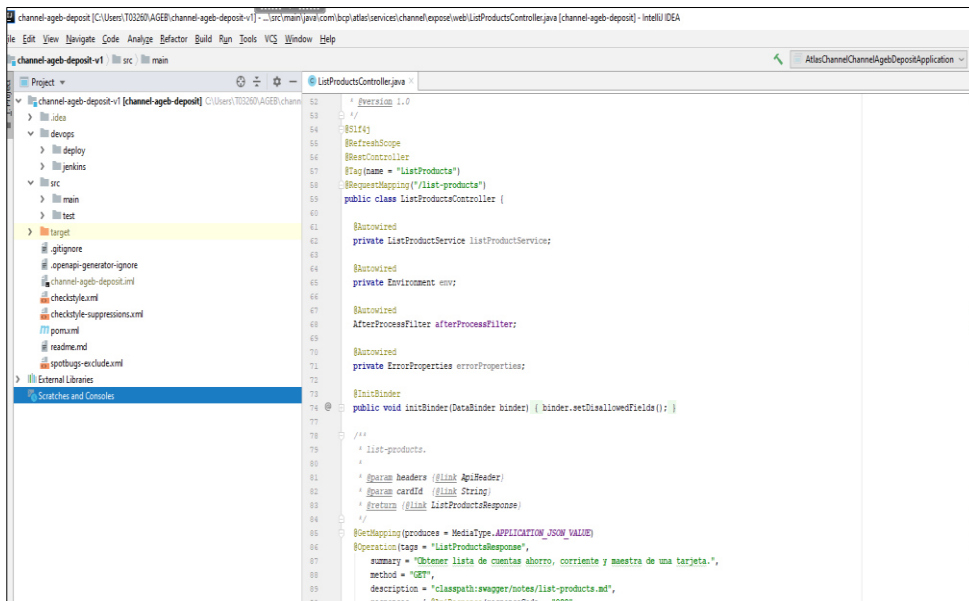
A grid of checkboxes for selecting dependencies:

- Hazelcast
- Soap Client
- Message Hub
- Event driven
- Rest Client
- Auditoria
- Redis
- Sql Server

Nota: Tomada de la Entidad Financiera.

El proyecto base generado es un archivo con extensión zip. A continuación, se realiza la importación del proyecto base utilizando el IDE IntelliJ IDEA y luego se procede a la codificación de la lógica de la API. Para el desarrollo de la API de depósito se debió seguir los lineamientos de desarrollo de la entidad financiera tales como el cumplimiento de porcentaje de cobertura de código, cumplimiento de checkstyles, spotbugs, etc.

Figura 15 : Codificación del API utilizando IntelliJ IDEA



Nota: Tomada de la Entidad Financiera.

Finalizado el desarrollo del API se procede al despliegue en el ambiente desarrollo, para esto se creó un Job con la herramienta Jenkins el cual es una herramienta para las tareas de integración continua, así como los procesos de despliegue en los distintos ambientes.

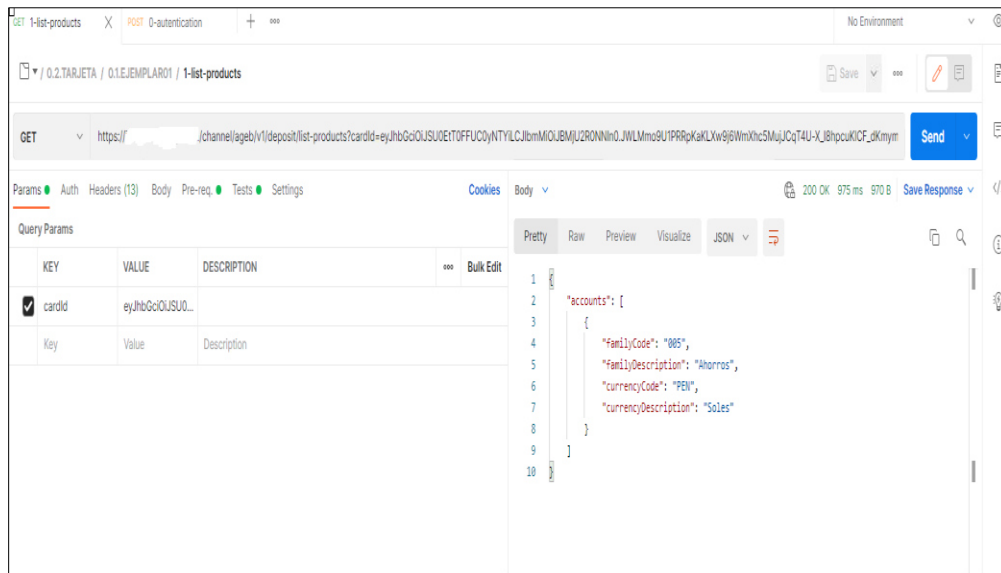
Figura 16 : Despliegue del API en el ambiente de desarrollo



Nota: Tomada de la Entidad Financiera.

Por último, se realizaron las pruebas con la herramienta POSTMAN en primer lugar y en segundo lugar se realizó la integración con el frontend.

Figura 17 : Pruebas a nivel de POSTMAN



Nota: Tomada de la Entidad Financiera.

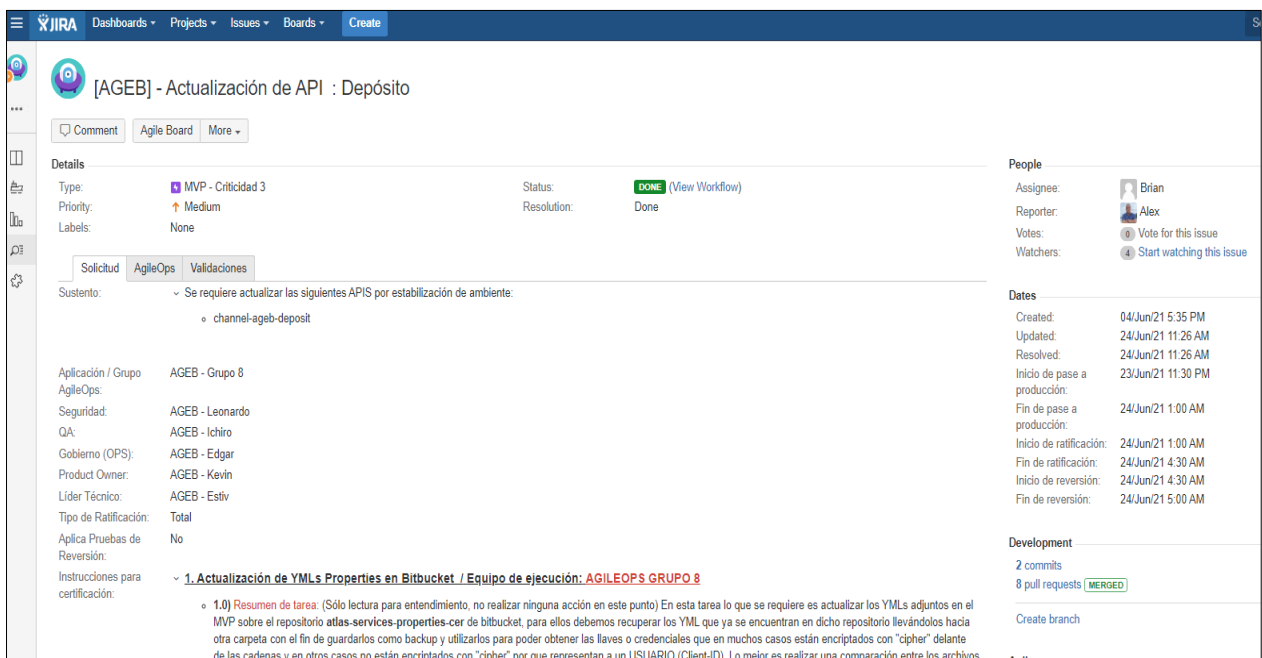
3.2.5.3 Pase a producción

Luego de que las pruebas resultaron satisfactorias en el ambiente de desarrollo y de haber cumplido con los lineamientos de seguridad de la entidad financiera se procede a realizar el pase a producción de la nueva funcionalidad. Para esto se debe de crear el ticket correspondiente utilizando la herramienta Jira.

El ticket describe los pasos a seguir para el despliegue del API en el ambiente de producción, así como también los equipos involucrados, los cuales pueden ser el equipo de seguridad, accesos, mainframe, cloud, etc. Adicionalmente en el ticket se debe de adjuntar todos los archivos de configuración necesarios para el pase a producción, así como también todas las conformidades para la ejecución el mismo.

El encargado de gestionar todo este proceso de pase a producción es el AgileOps del squad. El AgileOps es la única persona que realiza los pases a producción, se encarga de validar todo el proceso, crear y asignar las tareas a los equipos involucrados y en caso se presente errores al momento de ejecutar el pase a producción y no se resuelvan en el tiempo estimado, entonces procede a la reversión del mismo.

Figura 18 : Ticket Jira para el pase a producción.



Nota: Tomada de la Entidad Financiera.

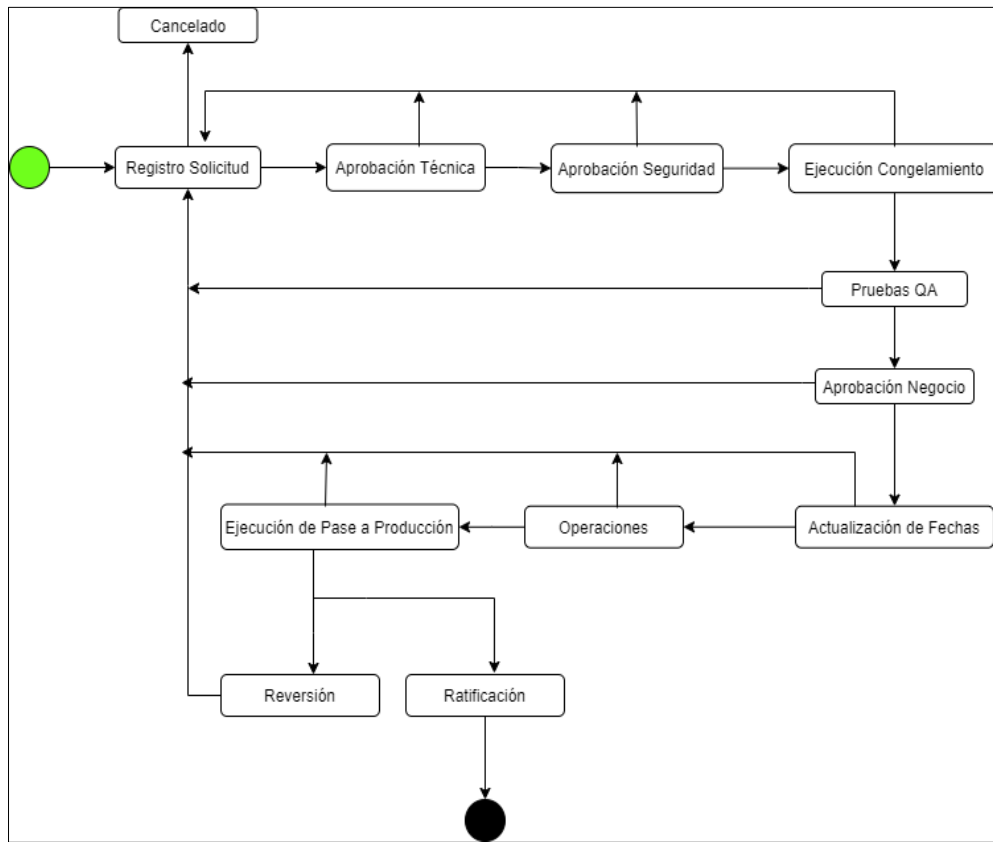
A continuación, se describe el flujo de un ticket Jira para un pase a producción:

- Registro Solicitud: Es el primer estado del ticket, se detallan los pasos o instrucciones a seguir para el despliegue del api en el ambiente de producción, además se adjunta todos los archivos de configuración necesarios.
- Aprobación Técnica: En este estado el líder técnico se encarga de verificar que los pasos a seguir y los archivos adjuntados se encuentren correctos.

Si se tienen observaciones en el ticket, entonces el ticket regresa al estado Registro Solicitud para su actualización.

- Aprobación Seguridad: El equipo de seguridad revisa las instrucciones y da su conformidad, caso contrario el ticket regresa al estado de Registro Solicitud.
- Ejecución Congelamiento: En estado el AgileOps es el encargado de ejecutar las instrucciones detalladas en el ticket, pero para el ambiente de certificación. Si se presentan observaciones se regresa el ticket al estado Registro Solicitud.
- Pruebas QA: El analista de calidad del squad se encarga de realizar las pruebas correspondientes en el ambiente de calidad, si todas las pruebas son satisfactorias entonces da su conformidad.
- Aprobación de Negocio: en este estado los especialistas de negocio de la entidad financiera son los encargados de dar su conformidad.
- Actualización de Fechas: el líder técnico se encarga del registro de la fecha y horas del pase a producción.
- Operaciones: el jefe de operaciones revisa el ticket y da su conformidad para el pase a producción.
- Ejecución de Pase a Producción: El AgileOps se encarga de ejecutar las instrucciones en el ambiente de producción, además se encarga de crear las subtareas y asignarlas a los otros equipos involucrados en el pase a producción.
- Reversión: el ticket llega a este estado cuando las pruebas realizadas por el analista de calidad no tienen resultados conformes, posteriormente el AgileOps ejecuta la reversión del pase a producción.
- Ratificación: En este estado el analista realiza las pruebas correspondientes, si todo es conforme se procede a dar terminado el ticket.
- Cancelado: cuando se decide cancelar un ticket por distintas razones.

Figura 19 : Flujo de un Ticket Jira.



Nota: Elaboración propia.

3.3 Evaluación

3.3.1 Beneficios Obtenidos

- Renovación de un total de 8000 terminales POS a nivel nacional.
- Incremento de las afiliaciones de nuevos agentes como resultado de la renovación de equipos y la realización de campañas de afiliación.
- Disminución de la tasa de error en la ejecución de las diferentes operaciones de los agentes.
- Al realizar la migración hacia una arquitectura de microservicios se pudo realizar la integración del backend con la herramienta de gestión de logs como Kibana , asimismo se realizó la integración con la herramienta Dynatrace para el monitoreo de la salud de los microservicios .
- La migración hacia la arquitectura de microservicios dio como resultado una alta escalabilidad y mantenibilidad del backend.
- Como resultado de la migración, el proceso de añadir nuevas transacciones a los terminales POS se podrán realizar sin mayor inconveniente.

CAPÍTULO IV – REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

El autor del presente trabajo de suficiencia participó en el proyecto de migración del backend del canal agente de una entidad financiera. El autor tuvo el rol de team member dentro del Squad Agentes que pertenece a la tribu de Canales alternativos.

La participación del autor fue en todas las etapas de migración del backend, tales como la creación del código fuente de las API's, pruebas en los ambientes de desarrollo y certificación, pase a producción de la transacción migrada. El marco de trabajo que se siguió para todo este proceso de migración fue Scrum, asimismo el autor formó parte de las distintas ceremonias tales como: Daily Scrum, Sprint Planning, Refinamiento, Retrospectiva y Sprint Review.

Toda esta experiencia ayudó al autor a aprender los lineamientos de desarrollo de APIs, el proceso de negocio del canal agentes y las políticas de seguridad de la entidad financiera. Asimismo, esto le permitió el aprendizaje de nuevas tecnologías tales como la programación reactiva y la nube de Azure, por otro lado, la experiencia de trabajar en un equipo ágil le permitió comprender y entender de mejor manera cómo funciona el marco de trabajo Scrum.

Como conclusión toda esta experiencia ayudará al autor a desenvolverse de una mejor manera en cualquier otro equipo de la entidad financiera en caso sea necesario, sumado a toda experiencia técnica y el prestigio que se ganó al ser parte de uno de los canales más importantes de la entidad financiera.

CAPÍTULO V – CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- El proceso de migración del backend del canal agentes el cual está compuesto por las transacciones de depósito, retiro, consulta de saldos y movimientos, emisión y descargos de giros, pago de servicios, pago fácil Sunat, transferencias se realizó de manera satisfactoria. Ver Anexo 2.
- Los retrasos que se presentaron en el proceso de migración fueron debidos a inestabilidad en los ambientes preproductivos (desarrollo, certificación) esto debido a la migración de otros componentes usados por el canal agentes, cambios en la infraestructura de la plataforma, cambios en los lineamientos de despliegue y demora en la gestión de pases a producción.
- El marco de trabajo que se empleó, en este caso Scrum ayudó a la entrega de valor al cliente, para esto el Product Owner del equipo realizó la priorización de las transacciones a migrar conjuntamente con los especialistas del negocio de la entidad financiera. Ver Anexo 3.
- Se realizó la integración del backend con el nuevo frontend desarrollado por un proveedor externo de la entidad financiera de manera satisfactoria.
- Se realizó la migración del parque de agentes por los nuevos terminales de manera gradual en los distintos agentes a nivel de Lima Metropolitana sin mayores contratiempos. Ver Anexo 4.
- Todo esto proceso de migración tuvo como resultado el aumento de afiliación de nuevos agentes ya que el contar con equipos de última generación y una interfaz amigable llamó la atención de los negocios esto sumado a los beneficios monetarios que brinda la entidad financiera. Ver Anexo 5.

5.2 Recomendaciones

Entre las recomendaciones más importantes tenemos las siguientes:

- Búsqueda de otro proveedor para el desarrollo de frontend ya que se presentaron muchos inconvenientes con el actual.
- Realización de capacitaciones en el uso del nuevo terminal POS a los agentes de manera supervisada.
- Dejar de usar por completo los componentes legados del banco.
- Realizar la implementación de BDD esto con el fin de describir a detalle el comportamiento de la aplicación (transacción) para que sea comprendida por todas las partes interesadas.
- A partir del presente trabajo de migración del backend del canal agente de una entidad financiera hacia una arquitectura de microservicios se podría considerar la aplicación de técnicas de machine learning para la mejorar la experiencia de los usuarios gracias al análisis del gran volumen de datos que genera la aplicación del canal.

5.3 Fuentes de Información

- Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices Architecture Enables DevOps: An Experience Report on Migration to a Cloud-Native Architecture. *IEEE*, 1-13.
- Dragoni, N., Giallorenzo, S., Lluch Lafuente, A., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). Microservices: yesterday, today, and tomorrow. *Springer*, 195–216.
- Financiera. (2021). *Via BCP*. Obtenido de Via BCP: <https://www.viabcp.com/nosotros>
- Microsoft. (2021). *Azure*. Obtenido de Azure: <https://azure.microsoft.com/en-us/overview/what-is-the-cloud/>
- Schmid, H. A. (1996). Creating applications from components: a manufacturing framework design. *IEEE*, 67-75.
- Schwaber , K., & Sutherland, J. (2016). *Scrum.org*. Obtenido de Scrum.org: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>
- Scrum.org*. (2021). Obtenido de Scrum.org: <https://www.scrum.org/resources/what-is-scrum>
- Stylos, J., Faulring, A., Yang, Z., & Myers, B. (2009). Improving API Documentation Using API Usage Information. *IEEE*, 119-126.

GLOSARIO

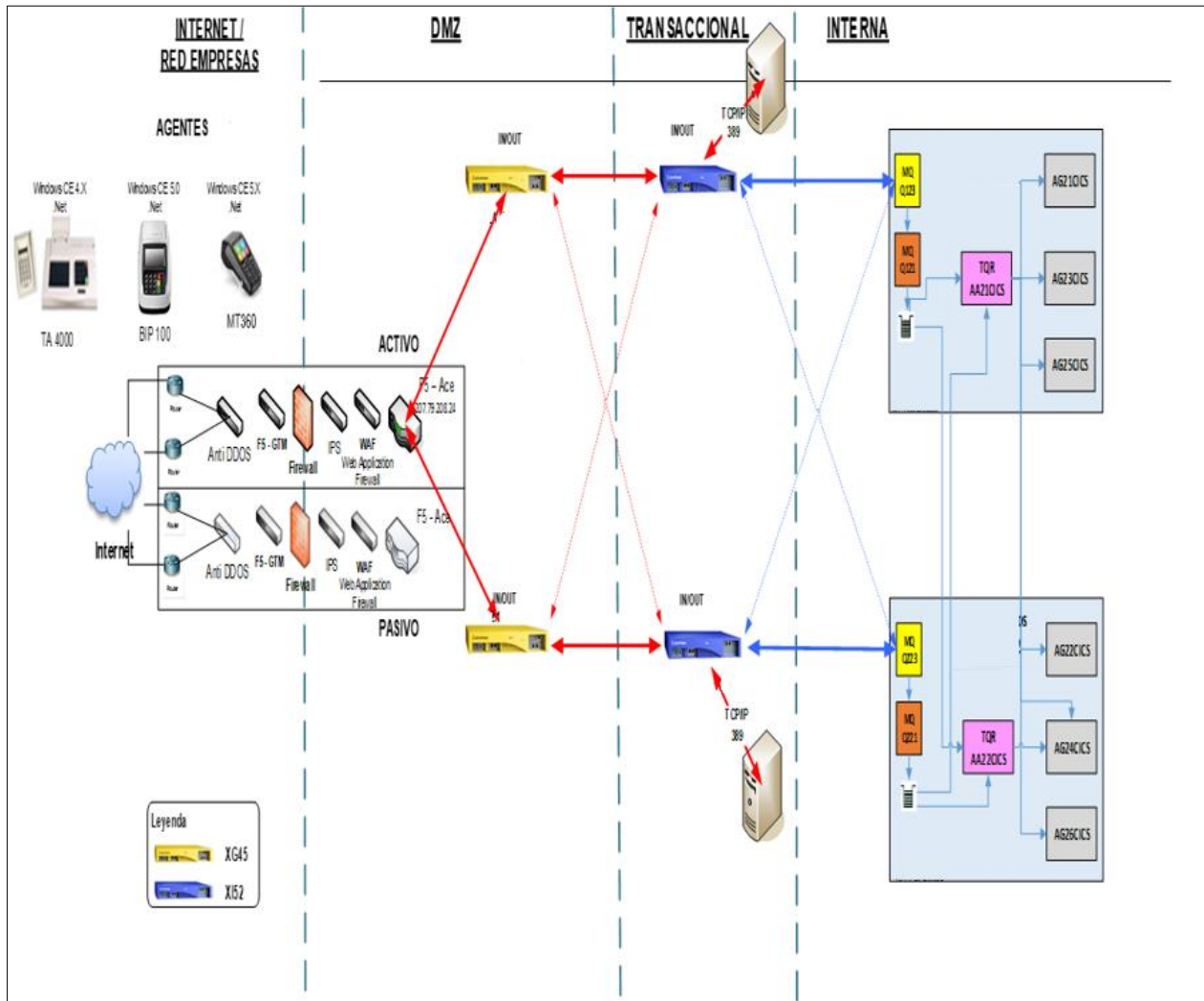
- AgileOps: persona que se encarga de gestionar y ejecutar los pases a producción.
- Agentes: personas jurídicas que se afilian a la entidad financiera.
- APK: son los archivos ejecutables desarrollados para Android.
- BDD: Son las siglas de Behavior Driven Development, es una estrategia de desarrollo guiado por comportamiento en el cual las pruebas están centradas en el usuario y el comportamiento del sistema.
- Checkstyle: herramienta para el análisis de código estático, comprueba si el código fuente cumple con las reglas de codificación.
- Contrato swagger: archivo yaml que contiene la descripción detallada del API, así como requisitos para ser consumida.
- Cross: componentes que contienen endpoints utilitarios para las otras API's.
- DataPower: realiza la transformación y el enrutamiento de mensajes como un bus de servicio empresarial.
- Endpoint: es un método o recurso único en la interfaz de nuestro servicio.
- Gobierno de APIs: equipo encargado de hacer cumplir los lineamientos de desarrollo y arquitectura relacionado a las APIs.
- Host: nombre por lo que se le conoce al mainframe.
- Mainframe: servidor usado por la entidad financiera por manejar entradas y salidas de gran volumen y enfatizar la computación de rendimiento.
- POS: terminal de venta usado por los agentes.
- SAAS: modelo de servicio que ofrece a los usuarios la posibilidad de conectarse a aplicaciones alojadas en la nube a través de Internet.
- Squad: equipo multidisciplinario y autónomo que trabajan bajo un marco.
- Spotbugs: es un programa que utiliza análisis estático para buscar errores en el código fuente.

- SwaggerHub: permite definir servicios web o APIs utilizando la especificación OpenAPI.
- Tribu: similar a un área de una entidad financiera con la diferencia que está comprendida por un grupo de squads.

ANEXOS

ANEXO 1. Arquitectura AS IS Detallada

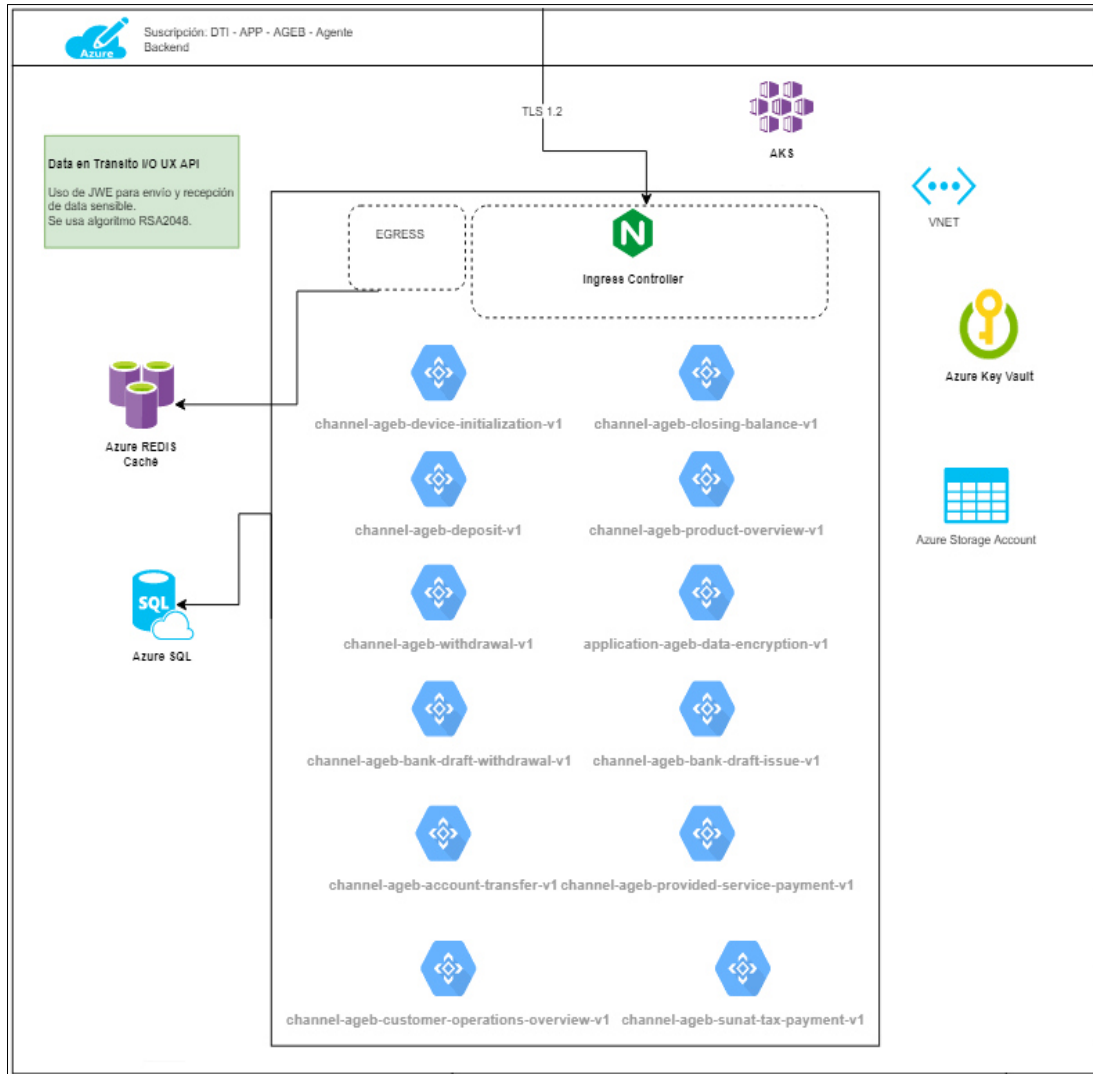
Figura 20 : Arquitectura AS IS detallado



Nota: Tomada de la Entidad Financiera.

ANEXO 2. Microservicios desarrollados

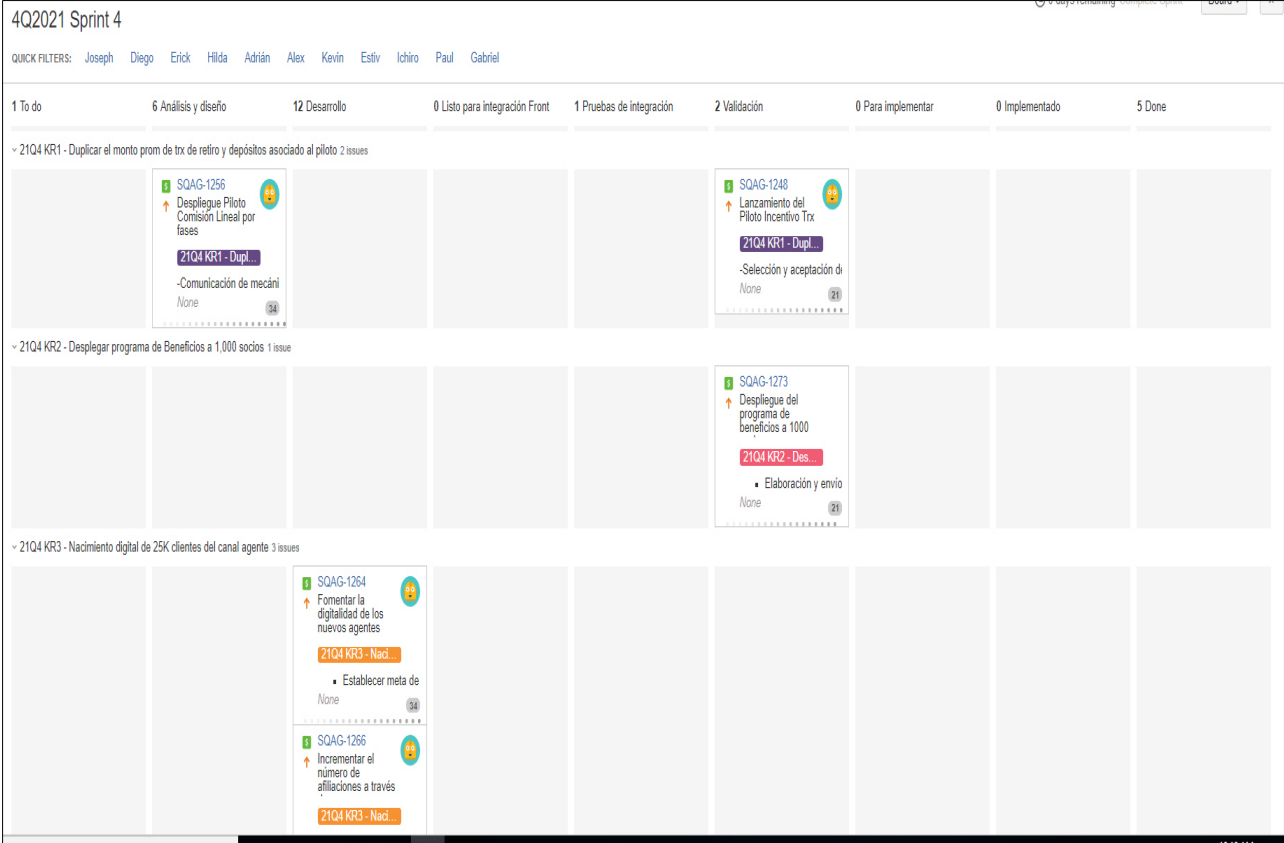
Figura 21 :Microservicios desplegados en la suscripción Azure



Nota: Tomada de la Entidad Financiera.

ANEXO 3. Historias de usuario de un Sprint

Figura 22 : Tablero Kanban



Nota: Tomada de la Entidad Financiera.

ANEXO 4. Renovación de los terminales POS

Figura 23 : Renovación del parque de agentes



Nota: Tomada de la Entidad Financiera.

ANEXO 5. Afiliaciones de nuevos agentes

Figura 24 : Diagrama de barras de afiliaciones de agentes



Nota: Tomada de la Entidad Financiera.