



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

**Modelo de integración para pago de servicios a través
de una entidad financiera**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniería de Sistemas

AUTOR

Andrea Alejandra ESPARRÓN MARTTI

ASESOR

Jorge Luis CHÁVEZ SOTO

Lima, Perú

2024



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

Esparrón, A. (2024). *Modelo de integración para pago de servicios a través de una entidad financiera*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas] Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Andrea Alejandra Esparrón Martti
Tipo de documento de identidad	DNI
Número de documento de identidad	47215146
URL de ORCID	https://orcid.org/0009-0009-8388-3219
Datos de asesor	
Nombres y apellidos	Jorge Luis Chávez Soto
Tipo de documento de identidad	DNI
Número de documento de identidad	08675814
URL de ORCID	https://orcid.org/0000-0002-9408-9266
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	John Ledgard Trujillo Trejo
Tipo de documento	DNI
Número de documento de identidad	06187585
Miembro del jurado 1	
Nombres y apellidos	Víctor Hugo Bustamante Olivera
Tipo de documento	DNI
Número de documento de identidad	25655590
Datos de investigación	
Línea de investigación	No aplica
Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento

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	2023
URL de disciplinas OCDE	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 15:55 horas del día 01 de febrero del año 2024, se reunieron virtualmente los docentes designados como Miembros del Jurado del Trabajo de Suficiencia Profesional, presidido por el Lic. Trujillo Trejo John Ledgard (Presidente), Lic. Bustamante Olivera Víctor Hugo (Miembro) y el Lic. Chávez Soto Jorge Luis (Miembro Asesor), usando la plataforma Meet (<https://meet.google.com/byr-ntdg-dks?hs=224>), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: **“MODELO DE INTEGRACIÓN PARA PAGO DE SERVICIOS A TRAVÉS DE UNA ENTIDAD FINANCIERA”**, por la Bachiller **Esparrón Martti Andrea Alejandra**; para obtener el Título Profesional de Ingeniería de Sistemas.

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

La 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 habiéndolo efectuado la calificación correspondiente por los miembros del Jurado, la Bachiller obtuvo la nota de **17 (DIECISIETE)**.

A continuación el Presidente de Jurados el Lic. Trujillo Trejo John Ledgard, declara a la Bachiller **Ingeniería de Sistemas**.

Siendo las 16:55 horas, se levantó la sesión.

Presidente

Lic. Trujillo Trejo John Ledgard

Miembro

Lic. Bustamante Olivera Víctor Hugo

Miembro Asesor

Lic. Chávez Soto Jorge Luis



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Vicerrectorado de Investigación y Posgrado



CERTIFICADO DE SIMILITUD

Yo JORGE LUIS CHÁVEZ SOTO en mi condición de asesor acreditado con la Resolución Decanal N° 001246-2023-D-FISI y Oficio N° 0010-PTTSP-FISI-2023 del Trabajo de Suficiencia Profesional cuyo título es MODELO DE INTEGRACIÓN PARA PAGO DE SERVICIOS A TRAVÉS DE UNA ENTIDAD FINANCIERA, presentado por el bachiller ESPARRÓN MARTTI ANDREA ALEJANDRA para optar el título de Ingeniero de Sistemas CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante el software de similitud textual, el documento evaluado cuenta con el porcentaje de 19% de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional**.

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del título correspondiente.

Firma del Asesor _____

DNI: 08675814

Nombres y apellidos del asesor: JORGE LUIS CHÁVEZ SOTO



DEDICATORIA

A mi esposo, hijas, padres y
hermanos por ser el principal
soporte y fortaleza a lo largo de mi
vida.

AGRADECIMIENTOS

Gracias a la Universidad Nacional Mayor de San Marcos, a la Escuela Profesional de Ingeniería de Sistemas y a todos mis estimados profesores en las diferentes etapas de mi vida por su valioso aporte a mi formación académica y profesional, así como de manera especial a mi asesor Jorge Chávez por su importante apoyo en el presente trabajo.

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

Modelo de Integración para pago de servicios a través de una entidad financiera

Autor: Esparron Martti, Andrea Alejandra

Asesor: Chávez Soto, Jorge Luis

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

Fecha: noviembre de 2023

RESUMEN

El presente trabajo de suficiencia profesional describe la implementación de un esquema genérico para la integración de los sistemas de recaudación de una entidad financiera con las empresas recaudadoras utilizando el “IBM Integration Bus” como middleware.

Como resultado de este proyecto, la entidad financiera logró brindar la afiliación de forma gratuita a las empresas recaudadoras, reducir el tiempo y disminuir la complejidad técnica de la integración, lo que hizo que una mayor cantidad de empresas se puedan interconectar en menor tiempo, lo que se traduce en un mayor ingreso de dinero.

Palabras Clave: Integración de sistemas, pago de servicios, middleware, empresa financiera.

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

Integration model for payment of services through a financial institution

Author: Esparron Martti, Andrea Alejandra

Advisor: Chávez Soto, Jorge Luis

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

Date: November 2023

ABSTRACT

This work of professional sufficiency describes the implementation of a generic scheme for the integration of the financial entity collection systems with the collection companies using the "IBM Integration Bus" as middleware.

As a result of this project, the financial entity was able to provide affiliation free of charge to the collecting companies, reduce the time and cost of affiliation, reduce the technical complexity of the integration, which made it possible to affiliate a greater number of companies in less time, which translates into a greater income of money.

Keywords: Systems integration, payment of services, middleware, financial company.

ÍNDICE DE CONTENIDO

Tabla de contenido	
RESUMEN	v
ABSTRACT	vi
ÍNDICE DE CONTENIDO	vii
ÍNDICE DE TABLAS	viii
ÍNDICE DE FIGURAS	ix
INDICE DE ANEXOS	x
INTRODUCCIÓN	1
CAPÍTULO I – TRAYECTORIA PROFESIONAL	3
CAPITULO II – CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA	6
2.1 Empresa – Actividad que realiza	6
2.2 Visión	6
2.3 Misión	6
2.4 Organización de la empresa	7
2.5 Área, cargo y funciones desempeñadas	8
2.6 Experiencia profesional realizada en la organización	8
CAPITULO III – ACTIVIDADES A DESARROLLAR	9
3.1 Situación problemática	9
3.1.1 Definición del problema	10
3.2 Solución	11
3.2.1 Objetivos	11
3.2.4 Fundamentos utilizados	15
3.2.5 Implementación de las áreas, procesos, sistemas y buenas prácticas	27
3.3 Evaluación	40
3.3.1 Evaluación Económica / Evaluación costo-beneficio	40
3.3.2 Interpretación del VAN y del TIR	41
CAPITULO IV – REFLEXIÓN CRÍTICA DE LA EXPERIENCIA	42
CAPITULO V – CONCLUSIONES Y RECOMENDACIONES	43
5.1 Conclusiones	43
5.2 Recomendaciones	43
5.3 Fuentes de Información	44
5.4 Glosario	46
ANEXOS	47

ÍNDICE DE TABLAS

Tabla 1: Cursos de Especialización	3
Tabla 2: Experiencia profesional	4
Tabla 3: Elementos de Scrum adaptados al proyecto	13
Tabla 4: Lista de épicas	27
Tabla 5: Product Backlog	28
Tabla 6: Detalle Trama Solicitud Consulta Parcial	31
Tabla 7: Detalle Trama Respuesta Consulta Parcial	32
Tabla 8: Costo del desarrollo del proyecto	40
Tabla 9: Costos totales de inversión	40
Tabla 10: Ganancia proyectada en 1 trimestre	41
Tabla 11: Resultado de la evaluación	41

ÍNDICE DE FIGURAS

Figura 1: Organigrama de la empresa.....	7
Figura 2: Integración basada en transferencia de archivos.....	10
Figura 3: Etapas de desarrollo de Software	12
Figura 4: Tablero Jira	14
Figura 5: Diferente cantidad de lógica encapsulada en un servicio	18
Figura 6: Interacción entre servicios, el servicio A usa al servicio B	19
Figura 7: Comunicación entre servicios	20
Figura 8: Valores en JSON.....	22
Figura 9: Object en JSON.....	23
Figura 10: Array en JSON	23
Figura 11: Number en JSON	23
Figura 12: Bus de integración.....	25
Figura 13: Arquitectura física del sistema.....	26
Figura 14: Diagrama de flujo funcional de Recaudación	29
Figura 15: Diagrama de componentes	30
Figura 16: Diagrama de Arquitectura de Integración.....	31
Figura 17: Flujo de integración RUTEADOR.....	33
Figura 18: Flujo de integración GENERICO	34
Figura 19: Flujo manejo ERROR	35
Figura 20: Repositorio Bitbucket usado para el código del proyecto.....	36
Figura 21: Ticket de Cambio para ejecutar el congelamiento	37
Figura 22: Web Administrativa de Interconexión	38
Figura 23: Web Administrativa de Interconexión	38
Figura 24: Aplicación de aprobaciones	39

INDICE DE ANEXOS

ANEXO 1: DISEÑO DEL MENSAJE JSON	47
---	-----------

INTRODUCCIÓN

Durante años las empresas recaudadoras se integraban a la entidad financiera sobre una solución basada en transferencia de archivos, pero con el aumento de la demanda de las empresas y usuarios con necesidad de utilizar este servicio se fueron generando diversos problemas: cada vez los archivos eran más grandes, se necesitó afiliar más empresas y los medios tecnológicos no permitían una transferencia de la data con seguridad o integridad lo que al final significaba archivos incompletos o ilegibles, esto causaba que la recaudación, para una o más empresas recaudadoras, se interrumpiera, en ocasiones por varios días, lo que al final se traduce en pérdida de dinero para ambas partes. Como un primer paso para mejorar el proceso de recaudación, algunas empresas empezaron a integrarse utilizando una solución ad hoc: se preparaban componentes de software específicos para cada empresa recaudadora, esto generaba un alto grado de complejidad técnica y alto tiempo de implementación, bajo este esquema era imposible masificar la oferta del servicio de recaudación.

Con el proceso de transformación digital, la organización solicitó generar una solución genérica de integración que permita agilizar el proceso de afiliación de nuevas empresas, así como también soportar múltiples y nuevos canales, como la billetera digital.

En la implementación se utilizó un middleware como integrador entre el sistema de recaudación de la entidad financiera y los servicios expuestos por las empresas recaudadoras, lo cual ayudó a que las nuevas afiliaciones no requieran desarrollos sino más bien configuración, haciendo que se requiera un menor tiempo de implementación, así como una menor complejidad técnica.

En el CAPÍTULO I se detalla la experiencia profesional de la autora, además de los diferentes cursos de especialización realizados a lo largo de su experiencia profesional, roles que desempeñó, así como las funciones y actividades realizadas en cada uno de los trabajos.

En el CAPÍTULO II se describe la organización en la que se desarrolla el presente trabajo, las actividades que se realizaron, así mismo la visión, la misión, organización de la empresa, área donde se participó y las funciones desempeñadas.

En el CAPÍTULO III se explica el problema principal, la solución, objetivo general y específicos, los alcances, las etapas, la metodología aplicada, fundamentos

utilizados e implementación de la solución. Del mismo modo, se detalló la evaluación económica que demostró la viabilidad del proyecto.

El CAPÍTULO IV se realizará una reflexión crítica sobre la experiencia profesional dentro del proyecto, las enseñanzas que se lograron, algunas deficiencias y los principales desafíos que se presentaron y que se tuvieron que superar para lograr el éxito en el proyecto.

En el CAPÍTULO V se mencionan conclusiones del autor y las recomendaciones de mejoras que se pueden implementar.

CAPÍTULO I – TRAYECTORIA PROFESIONAL

1.1 Presentación profesional

La autora del presente trabajo es Bachiller en Ingeniería de Sistemas de la Universidad Nacional Mayor de San Marcos con más de 7 años de experiencia en la Integración de Aplicaciones en los rubros de Educación, Banca y Seguros, integrando aplicaciones con diversas herramientas de IBM tales como IBM WebSphere Message Broker, IBM Integration Bus e IBM WebSphere MQ.

Durante su trayectoria profesional ha demostrado sus capacidades de trabajo en equipo, comunicación, adaptación, pensamiento crítico y creatividad. Se ha destacado en los diversos proyectos en los que ha participado, teniendo siempre opiniones positivas de su desempeño.

La autora ha realizado diversos cursos de especialización que la han llevado a poder implementar con éxito los proyectos que tuvo a su cargo, así como también ha compartido estos conocimientos en diversas capacitaciones que ha dictado. En la Tabla 1, se indican los cursos realizados.

Tabla 1: Cursos de Especialización

Curso	Institución	Fecha
IBM Integration Bus V9	IBM Education Training	2016
IBM WebSphere MQ V8	IBM Education Training	2016
Scrum	Instituto GESAP	2018

Nota: Fuente. Elaboración propia

1.2 Experiencia Profesional

La tabla 2 detalla la experiencia profesional de la autora del presente trabajo.

Tabla 2: Experiencia profesional

NTT Data	
Duración	2022 – Actualidad
Cargo	Líder Técnico
Funciones	Realizar el análisis, diseño y desarrollo de las soluciones de integración. Guiar al equipo de desarrollo en las decisiones técnicas del proyecto. Liderar las reuniones técnicas. Evaluar los objetivos de los proyectos y esquematizar la mejor solución. Realizar la gestión para la afiliación de las nuevas empresas.
Synopsis	
Duración	2019 – 2022
Cargo	Analista de Integración
Funciones	Mantenimiento de las API's del cliente, usando Oracle, NET 6, API Management en Azure. Analizar, documentar y desarrollar requerimiento de interfaces del ERP Banner para exponer endpoints a través de las APIS. Análisis, diseño, implementación y pruebas de flujos de integración en IBM Integration Bus.
MDP Consulting	
Duración	Febrero, 2015 – 2018
Cargo	Analista Técnico de Integración
Cliente	La Positiva, Sura, Yanbal

Funciones	Análisis de los requerimientos expresados por los proyectos. Estimar esfuerzo para la atención de los requerimientos. Proponer y documentar el diseño de la solución. Implementar la solución de integración y realizar pruebas unitarias. Apoyar en las pruebas integrales de la aplicación. Corregir bugs detectados durante el ciclo de vida del proyecto. Desarrollo del Documento de Análisis Funcional y Técnico. Definición y diseño de interfaces de servicios usando XSD, WSDL. Implementación de proyectos de integración con SAP. Conexión a base de datos DB2 y Oracle desde los flujos de integración. Manejo de errores y uso de patrones de integración.
------------------	--

Nota: Fuente. Elaboración propia

CAPITULO II – CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA

2.1 Empresa – Actividad que realiza

La organización, objeto del presente trabajo, es una empresa líder del Perú, cuenta con diversos canales de atención tanto físicos y digitales, esto hace que la mayoría de las empresas recaudadoras quieran afiliarse a su sistema de recaudación.

Es en este contexto, que la empresa decide implementar una solución genérica que permita que la afiliación sea más rápida y eficiente.

2.2 Visión

“Servir de la mejor manera al cliente”.

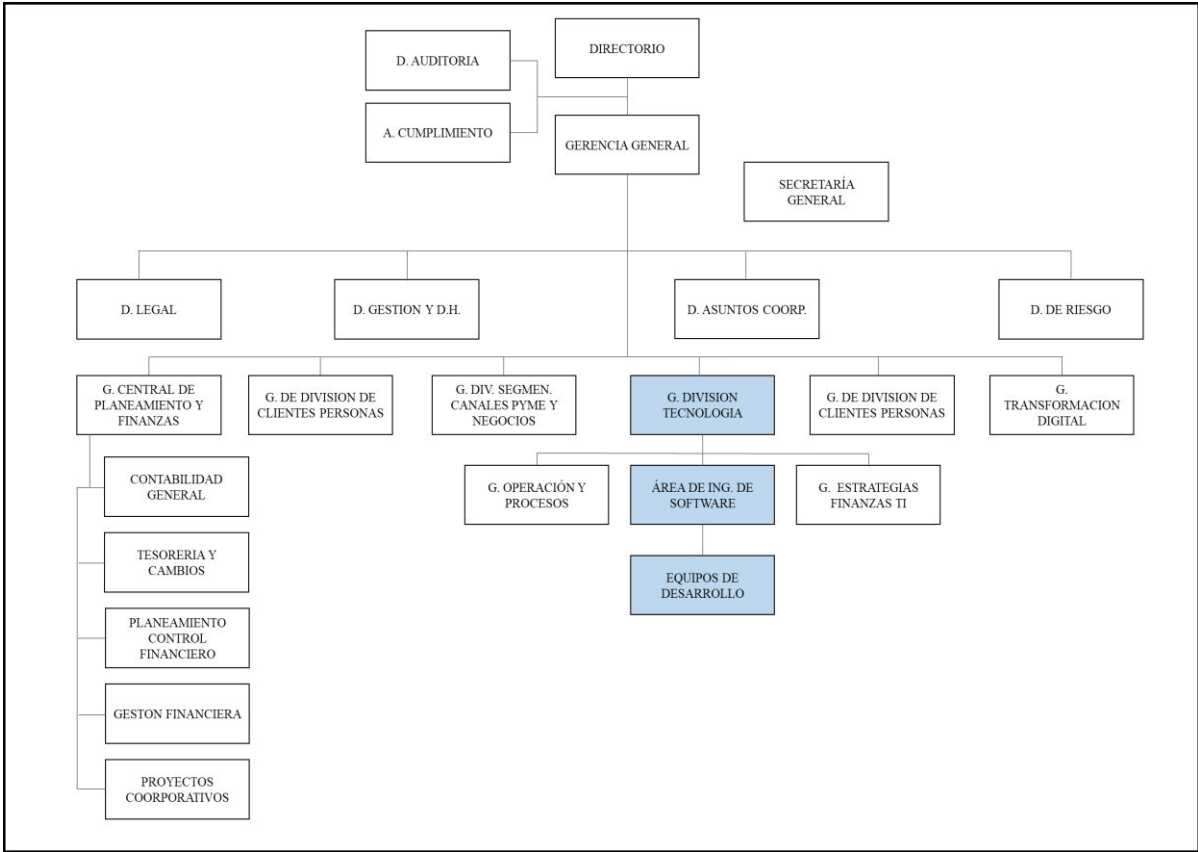
2.3 Misión

“Ser una entidad financiera simple, transaccional, rentable y con personal altamente capacitado y motivado”.

2.4 Organización de la empresa

En la figura número 1, se puede visualizar el organigrama general de la empresa y se ha resaltado en celeste las áreas involucradas en el presente trabajo de suficiencia profesional.

Figura 1: Organigrama de la empresa



Fuente. Elaboración propia, adaptado de la organización

2.5 Área, cargo y funciones desempeñadas

Los diferentes equipos se dividen de acuerdo con el tipo de productos y servicios con los que cuenta la entidad financiera, la autora del presente trabajo se ha desempeñado como líder técnica en el equipo encargado del sistema de recaudación.

Las funciones desempeñadas en el cargo han sido:

- Analizar los requerimientos.
- Diseñar los flujos de integración de acuerdo con los estándares y lineamientos que maneja la entidad financiera.
- Participar en reuniones técnicas con las empresas que se afiliarán a la recaudación genérica.
- Definir la estrategia que se deben seguir para realizar la integración de las empresas recaudadoras.
- Hacer seguimiento de las empresas interconectadas al sistema de recaudación.
- Capacitar a nuevos desarrolladores en las herramientas, framework y aplicaciones de la entidad financiera.

2.6 Experiencia profesional realizada en la organización

Durante la experiencia profesional en la entidad financiera, la autora se ha desempeñado como líder técnico, capacitando y guiando a los equipos de desarrollo en los proyectos realizados.

Las funciones desempeñadas en la organización fueron las siguientes:

- Analizar, diseñar y definir soluciones de integración.
- Diseñar soluciones de integración alineadas con los requerimientos.
- Diseñar la arquitectura de la integración en base al middleware IBM Integration Bus.
- Asegurar la calidad de desarrollo de los flujos del equipo de desarrollo.

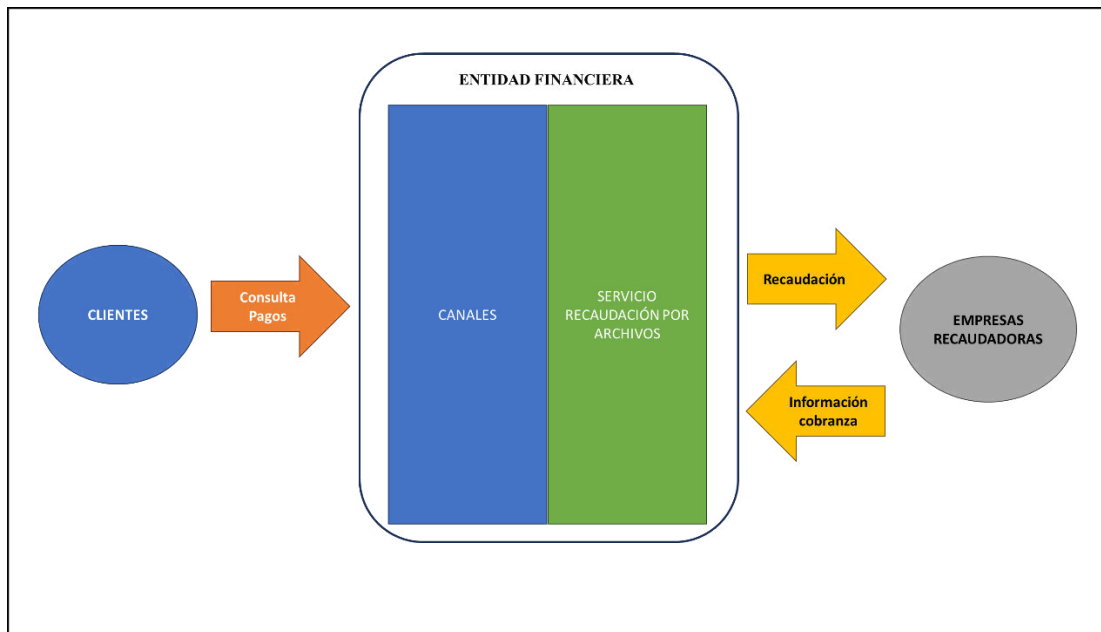
CAPITULO III – ACTIVIDADES A DESARROLLAR

3.1 Situación problemática

En nuestro país existen diversas empresas que ofrecen servicios tales como luz, agua, teléfono, educación, internet, entre otros. Originalmente los clientes debían acercarse a las oficinas de estas empresas para poder hacer el pago de sus servicios, lo que implicaba pérdida de tiempo, largas colas, costo en transporte, esto hacía que las empresas no pudiesen masificar sus servicios ya que no se daban abasto para la atención, en sus oficinas, de los clientes en el proceso de recaudación. Es en este contexto, que las empresas buscaron alianzas con las entidades financieras, ya que estas contaban con muchas oficinas y una red de atención a nivel nacional lo que les permitía un nivel de cobertura mayor.

Inicialmente, la entidad financiera contaba con una solución basada en transferencia de archivos, como se muestra en la Figura 2. Las empresas recaudadoras debían enviar su archivo de cobranza donde se encontraban los datos de sus clientes y sus deudas, los clientes utilizaban los diferentes canales de la entidad financiera para poder realizar las consultas y pagos de sus servicios, la entidad financiera ejecutaba la recaudación en todo el día y al finalizar ese día se ejecutaba el proceso de conciliación; al día siguiente reportaba los resultados a la empresa. Esta solución supone varios problemas, en el día no se podía actualizar las deudas y el pago demoraba en verse reflejado en la empresa recaudadora mínimo dos días. La propia transferencia de estos archivos generaba retos para ambas instituciones tales como no contar con un mecanismo confiable de transferencia de datos sensibles, los archivos eran muy grandes y podían llegar con data corrupta y esto generaba no poder recaudar varios días.

Figura 2: Integración basada en transferencia de archivos



Fuente. Elaboración propia

3.1.1 Definición del problema

El principal problema era que no existía una solución genérica y masiva de integración que permitiera que las empresas se puedan afiliar al sistema de recaudación de forma rápida y fácil, además de contar con la información de su cobranza en línea.

3.2 Solución

Implementación de una solución genérica y masiva de integración que permita la fácil afiliación de las empresas recaudadoras a la entidad financiera y que les permita contar con la información en línea de los pagos realizados.

3.2.1 Objetivos

3.2.1.1 Objetivo general

Implementar una solución genérica y masiva de integración, entre la entidad financiera y empresas recaudadoras usando middleware para la recaudación en línea.

3.2.1.2 Objetivos específicos

- Analizar el proceso funcional de la recaudación y los escenarios de comunicación entre la entidad financiera y empresas recaudadoras.
- Analizar e identificar los componentes involucrados en la recaudación.
- Diseñar la integración genérica en línea entre la entidad financiera y empresas recaudadoras.
- Desarrollar y desplegar la integración genérica en línea entre la entidad financiera y empresas recaudadoras.
- Permitir la afiliación sencilla de las empresas recaudadoras, a través de la implementación de formatos genéricos.

3.2.2 Alcance

Alcance funcional

El alcance del presente trabajo es realizar el análisis, diseño y desarrollo de la integración entre la entidad financiera y las empresas recaudadoras mediante IBM Integration Bus.

El presente proyecto cubre los procesos de recaudación que consta de los servicios de consulta, pago y extorno, y está basado en las modalidades parcial y completa.

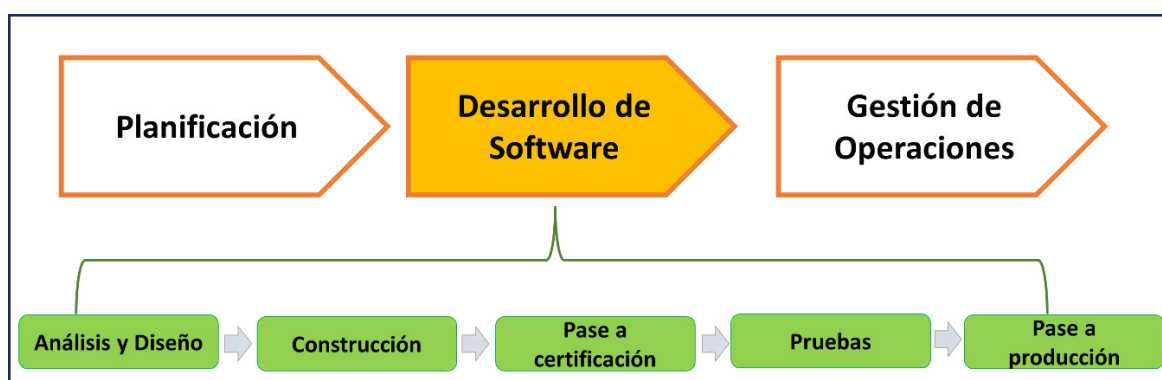
Alcance organizacional

El alcance organizacional comprende el área de recaudación de la entidad financiera.

3.2.3 Etapas y Metodología

El desarrollo de software del proyecto se divide en las siguientes cinco etapas que se ven en la figura número 3:

Figura 3: Etapas de desarrollo de Software



Fuente. Elaboración propia adaptado de la empresa

3.2.3.1 Etapa 1: Análisis y Diseño

El alcance de esta etapa es, desde el requerimiento priorizado por el negocio hasta la elaboración de los diagramas de arquitectura de la solución. Para esto, el arquitecto de integración junto al líder técnico, elaboran el diagrama de contexto funcional, diagrama de contexto técnico y diagrama de componentes e integración. Proceden a validar los lineamientos de seguridad y de Datos con cada especialista de estas áreas. Estos diagramas son alojados en el repositorio de la entidad financiera.

Con el diagrama de arquitectura de solución técnica validado, el líder técnico se encarga de realizar el diseño detallado de la solución.

3.2.3.2 Etapa 2: Construcción

La metodología de trabajo del equipo de desarrollo es SCRUM. En la tabla 3 se muestran los componentes que se utilizan:

Tabla 3: Elementos de Scrum adaptados al proyecto

COMPONENTES	DESCRIPCION
SPRINT PLANNING	• Este evento se realiza al inicio del sprint. • Se establecen metas para el sprint. • Escogen historias de usuario que soporten el objetivo. • Desglosan historias en tareas específicas. • Se define el Sprint Backlog.
DAILY SCRUM	• Inspección del tablero • Sincronización del equipo • Plan de trabajo del día
SPRINT REVIEW	• El equipo liderado por el Product Owner presenta lo que lograron durante el Sprint a stakeholders. • El equipo de desarrollo habla sobre lo que estuvo bien, los problemas que tuvieron y cómo los resolvieron.
SPRINT RETROSPECTIVE	• Se realiza al final del sprint, tras el sprint review y antes del siguiente planning. • Se inspecciona como fue el sprint. • Se crea un plan para implementar mejoras.
DEFINITION OF READY (DoR)	• Lista de verificación de criterios por las cuales una historia se encuentra lista para ser implementada dentro de un sprint

DEFINITION OF DONE (DoD)

- Lista de verificación de actividades requeridas para tener un producto completo.

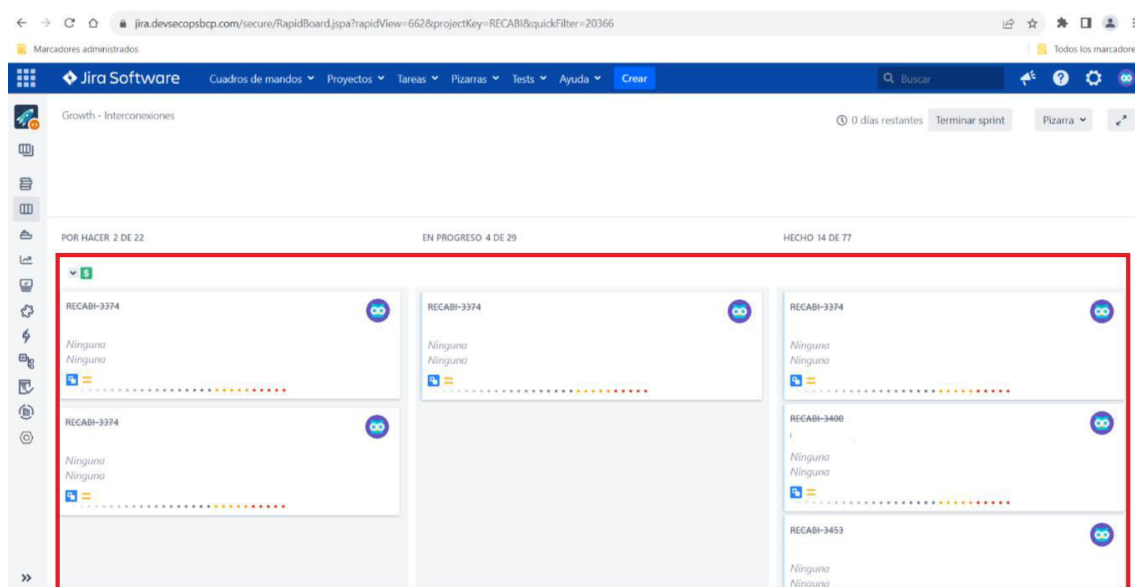
Nota. Fuente: Elaboración propia

Los procesos por seguir en el desarrollo de una nueva integración son:

- Codificar, realizar pruebas unitarias y revisión de pares.
- Analizar la calidad y seguridad del código.
- Proteger y empaquetar el código.
- Realizar pruebas de integración.
- Crear la solicitud del cambio que incluye todos los pasos que deben realizarse para instalar la solución tanto en certificación y producción, esto es validado por el Analista de calidad, Arquitecto de integración y Analista de Seguridad.

En la figura número 4 se muestra un ejemplo del tablero SCRUM, la zona resaltada en el cuadrado rojo identifica una historia de usuario, a su vez se divide en tareas que según su estado se ubican en las columnas “POR HACER”, “EN PROGRESO” o “HECHO”.

Figura 4: Tablero Jira



Fuente. Documentación del proyecto

3.2.3.3 Etapa 3: Pase a certificación

En esta etapa, el líder técnico se encarga de registrar la solicitud de congelamiento y realizar la validación de los requisitos según la lista de lineamientos requeridos. El equipo encargado de gestionar la instalación se encarga de derivar las tareas a los diferentes equipos técnicos involucrados, luego de que las tareas son realizadas se da por finalizada la instalación en el ambiente de certificación.

3.2.3.4 Etapa 4: Pruebas

En esta etapa se tiene la data de prueba preparada previamente y el analista de calidad se encarga de ejecutar los casos de prueba, que incluyen pruebas de rendimiento, funcionales y pruebas de seguridad. Luego que finalizan las pruebas, valida y reporta los resultados obtenidos, en caso de que todo sea correcto se continua con la siguiente etapa. En caso de error, se debe regresar a la etapa anterior y realizar los cambios necesarios para volver a ejecutar el congelamiento y nuevamente iniciar la etapa de pruebas.

3.2.3.5 Etapa 5: Pase a Producción

Luego de que las pruebas hayan sido satisfactorias, el líder técnico debe planificar el pase a producción en el calendario de pase, luego debe solicitar la aprobación de este cambio en el comité de pases, donde se realiza la validación del impacto en los diferentes sistemas de la entidad financiera. Con el pase planificado en el calendario y la aprobación del comité, se procede a ejecutar el pase a producción el día reservado, ese mismo día al finalizar la instalación, el analista de calidad se encarga de probar en el ambiente de producción y brindar su conformidad para dar por finalizado el pase a producción.

3.2.4 Fundamentos utilizados

3.2.4.1 Estilo de arquitectura de software.

Un estilo de arquitectura representa la recopilación de experiencias, buenas prácticas que los arquitectos pueden observar en busca de oportunidades de reusar y aprovechar esas experiencias. (Eeles y Cripps, 2010).

Adoptar un estilo de arquitectura de software ayuda a las organizaciones a aprovechar las mejores prácticas que llevaron a los arquitectos a obtener buenos

resultados y evitar errores o fallas que llevaron al fracaso esfuerzos de múltiples organizaciones, esto hace que esas experiencias sean invaluable para los arquitectos en sus nuevas iniciativas. (Len, Clements y Kazman, 2013)

Como ejemplo de estilos de arquitectura conocidos tenemos: Arquitectura Orientada a Servicios.

a) Arquitectura de Software

Kruchten (2000) Señala que una arquitectura de software es un conjunto de decisiones significativas acerca de la organización, de los sistemas de software, la selección de elementos estructurales y las interfaces con las que el sistema estará formado, junto con la funcionalidad especificada en la colaboración entre estos elementos, la propia composición de estos elementos en subsistemas cada vez más grandes y complejos y el estilo arquitectónico que guía a la organización.

b) Arquitectura Orientada a Servicios - SOA

Es un estilo de arquitectura donde la lógica requerida para resolver un problema grande es construido, ejecutado, gestionado, de forma más sencilla descomponiendo el problema en una colección de piezas más pequeñas. Cada pieza se encarga de soportar una parte específica de la funcionalidad o de parte del problema. (Erl, 2005)

Cada una de las piezas deberá tener una responsabilidad bien definida, interfaces claras y la mayor independencia posible de su entorno o de las demás piezas. A estas piezas se les denomina servicios.

Ventajas de SOA

Las empresas que adoptan SOA como estilo de arquitectura de software obtienen las siguientes ventajas:

- **Mayor agilidad del negocio; comercialización más rápida:**

Al tener un catálogo de servicios listos para ser utilizados se logrará implementar aplicaciones que soporten nuevas necesidades de negocio de forma rápida y a bajo costo. Ya no será necesario implementar las aplicaciones desde cero (IBM, 2022a). La respuesta al entorno cambiante, estrategias y objetivos de negocio será rápida y eficiente.

- **Capacidad para aprovechar la funcionalidad existente en nuevos mercados**

IBM (2022a) menciona que muchas empresas se encontraban con limitaciones tecnológicas que impedían el aprovechamiento de funcionalidad existente, por ejemplo: Entidades financieras que mantenían su core sobre tecnología Mainframe, lo que la hacía imposible exponer toda esa funcionalidad desde medios modernos como el WEB o aplicaciones en teléfonos. Pero usando la arquitectura SOA se puede exponer esta funcionalidad de forma transparente para diferentes canales y así aprovechar toda la funcionalidad existente, ahorrándole a las empresas mucho tiempo y dinero que emplearía en reescribir toda esa funcionalidad.

- **Mejora de la colaboración entre negocio y TI**

En una SOA los servicios se pueden definir en términos de funciones de negocio, por ejemplo: “Consultar Saldo”, “calcular ROI”. Esto permite al equipo del área de negocios poder interactuar de forma más efectiva con los equipos de desarrollo y lograr una mejor sinergia entre ellos, de forma que, logren alinear mejor los esfuerzos y recursos a los objetivos del negocio. (IBM, 2022a)

Componentes de SOA

Existen cinco componentes principales que forman parte de la arquitectura orientada a servicios:

- **Servicio**

Son componentes de la arquitectura orientada a servicios que encapsulan lógica, mantienen independencia del contexto en el que existen y presentan una interfaz bien definida, que le permite interactuar con su entorno de forma que pueda proveer dicha lógica. (Eeles, 2005)

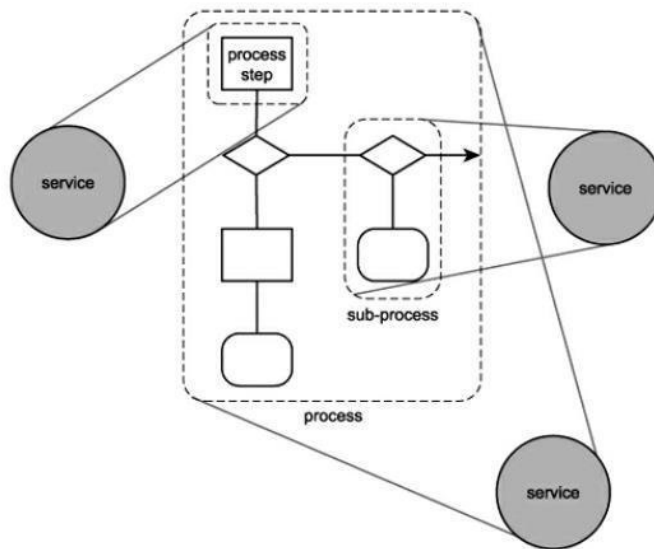
Cada servicio individual tiene tres características principales:

- **Implementación de servicios**

La implementación de servicios es el código que soporta la lógica para realizar la función de negocio que el servicio específico, como la autenticación de usuarios o el cálculo de una factura.

Los servicios como se ve en la figura 5, pueden encapsular una variada cantidad de lógica, esto dependerá de la necesidad y de la función de negocio a la que soporta. (Erl, 2005)

Figura 5: Diferente cantidad de lógica encapsulada en un servicio



Nota: Tomado de (Erl, 2005).

- **Contrato del servicio**

El contrato del servicio define su naturaleza, sus términos y condiciones asociados, como los prerequisites para utilizar el servicio, además puede incluir su costo y la calidad o los compromisos de calidad proporcionado como: tiempos de respuesta, volumetría máxima.

- **Interfaz del servicio**

En SOA, otros servicios o sistemas se comunican con un servicio a través de su interfaz. Esta interfaz define la manera en que se puede invocar al servicio. Las interfaces reducen las dependencias entre los servicios y los solicitantes. Al estar la interfaz bien definida se podrá aprovechar esta para usar el servicio sin importar la implementación subyacente, no es necesario saber en qué lenguaje de programación fue desarrollado. (Amazon, 2022)

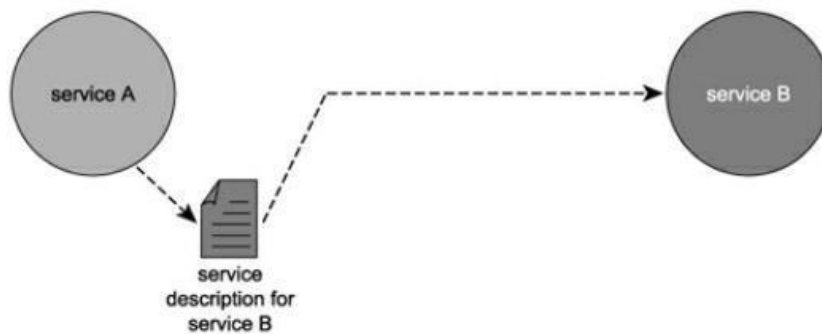
- **Interacción entre servicios**

Los servicios pueden ser usados por otros servicios o por programas, para lograr una efectiva interacción entre servicios sus interfaces como se muestra en la figura 6, lógica(comportamiento), dependencias, restricciones deberán ser definidas y expuestas con claridad, esto a través de una adecuada descripción del servicio.

Según señala Erl (2005), una descripción básica del servicio incluye:

- Nombre
- La data que recibe (mensaje de entrada)
- La data que retorna (mensaje de salida)
- Los eventos que genera

Figura 6: Interacción entre servicios, el servicio A usa al servicio B



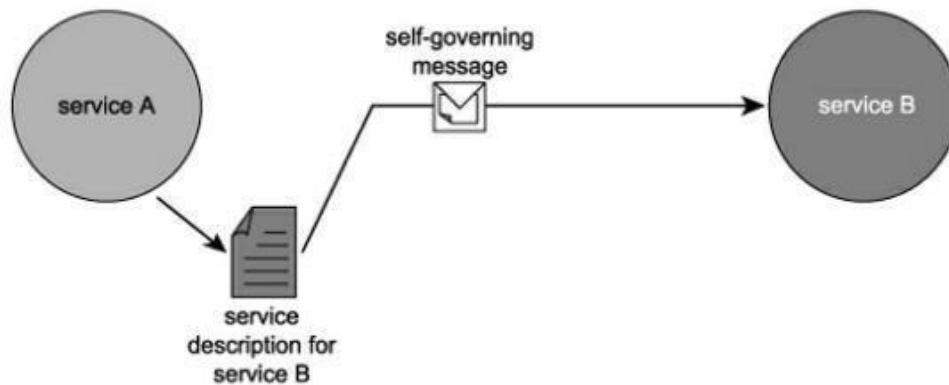
Nota: Tomado de (Erl, 2005).

- **Comunicación entre servicios**

La comunicación entre servicios se hace posible a través de mensajes como se ve en la figura 7, estos mensajes son enviados desde el consumidor hacia el proveedor del servicio, este deberá seguir las reglas impuestas y declaradas en la definición de la interfaz del servicio.

Para que los servicios interactúen es necesario un medio de comunicación que permita el intercambio de mensajes. En SOA este medio lo provee el ESB (Enterprise Service BUS).

Figura 7: Comunicación entre servicios



Nota: Tomado de (Erl, 2005).

- **Proveedor de servicios**

El proveedor de servicios define, crea, mantiene, documenta y proporciona uno o más servicios que otros pueden utilizar. Las organizaciones pueden crear sus propios servicios o adquirirlos de proveedores de servicios externos.

- **Consumidor de servicios**

El consumidor de servicios solicita al proveedor de estos poner en marcha un servicio específico. Puede ser un sistema completo, aplicación u otro servicio. El contrato de servicio especifica las reglas que el proveedor y el consumidor de servicios deben seguir al momento de interactuar entre sí. Los proveedores y consumidores de servicios pueden pertenecer a departamentos, organizaciones o incluso sectores diferentes.

- **Registro de servicios**

Un registro de servicios, o repositorio de servicios, es un directorio de servicios disponibles accesible a través de redes. Almacena documentos descriptivos sobre el servicio que pertenecen a los proveedores de servicios. Los documentos descriptivos contienen información acerca del servicio y cómo comunicarse con él. Los consumidores de servicios pueden descubrir fácilmente los servicios que necesitan por medio de dicho registro.

Enterprise Service BUS (ESB)

Este componente es responsable de la integración entre consumidores y proveedores de servicios.

Como parte de sus responsabilidades tiene:

Transformar modelos de datos, tanto consumidores como proveedores podrían tener interfaces y modelos de mensajes diferentes, es responsabilidad del ESB transformar estos modelos de datos del formato generado por el consumidor al que está definido por la interfaz del proveedor.

Algunos de los modelos de datos soportados son:

- **XML**

Extensible Markup Language (XML), describe una clase especial de objetos especializados en representar datos, estos objetos son llamados “Documentos XML”, además describe parcialmente el comportamiento de programas de software especializados en procesarlos (W3C, 2008).

- **Documentos XML**

Los documentos están compuestos por unidades llamadas entidades. Las entidades se pueden referir a otras entidades o a datos finales.

- **Procesador XML**

Es un tipo especial de programa de software que se encarga de leer, extraer los datos y hacer disponible tanto los datos y la estructura de los documentos XML a otros programas, que son los verdaderos interesados en el contenido de los documentos XML.

Los documentos XML pueden seguir la estructura definida en un documento llamado document type definition (DTD). Los procesadores pueden usar estos DTD para validar que la estructura de los documentos XML cumplan con las restricciones impuestas sobre ellos. (W3C, 2008)

- ✓ **Ventajas de XML**

IBM (2019-2022) señala como ventajas de los documentos XML las siguientes:

XML utiliza lenguaje humano, no informático. XML es legible y comprensible, incluso por principiantes.

XML es 100% portable. Cualquier aplicación que pueda procesar XML puede utilizar su información, independientemente de la plataforma.

XML es ampliable. Cree sus propias etiquetas, o utilice etiquetas creadas por otros, que utilicen el lenguaje natural de su dominio, que tengan los atributos que necesita y que tengan sentido para usted y sus usuarios.

- **JSON**

JavaScript Object Notation fue inspirado por los “object literal” de JavaScript, que si bien sigue la ECMA Script Language Specification (ECMA, 2017), no es de estricto cumplimiento, tampoco especifica completamente el intercambio de datos entre programas de software.

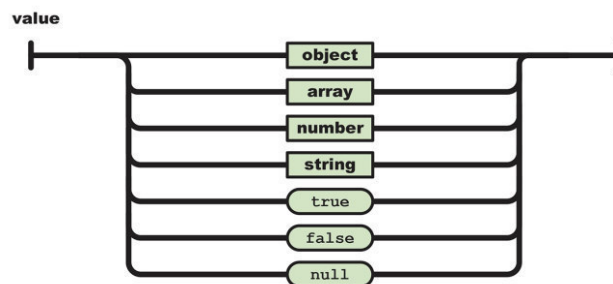
JSON define un marco de trabajo enfocado en la sintaxis del mensaje, pero no en su semántica, cada par proveedor-consumidor acordara la semántica asociada a los documentos JSON para cada uso particular. (ISO/IEC, 2017)

Un texto JSON es una secuencia de pares nombre-valor, ambos en Unicode.

- **Valores JSON**

Los valores en JSON pueden ser object, array, number, string, true, false, null, como se ve en la figura 8.

Figura 8: Valores en JSON



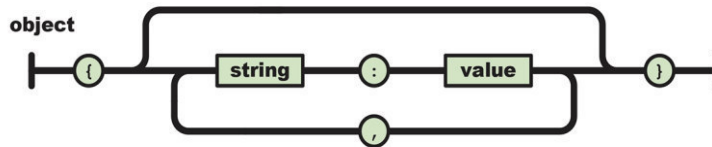
Nota: Tomado de (ISO/IEC, 2017)

- **Object**

Un Object en JSON es representado por un par de llaves, una izquierda y una derecha “{”. “}” y dentro de ellas cero o más pares nombre-valor separados por una coma “,”.

Se usa los dos puntos para separar los nombres de los valores. JSON no imponen ninguna restricción especial respecto a la secuencia de caracteres que forman el nombre.

Figura 9: Object en JSON

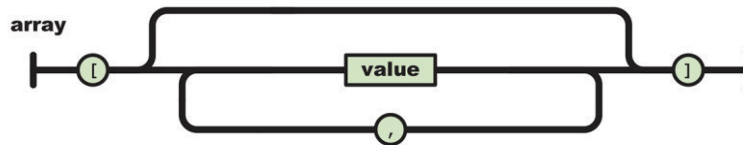


Nota: Tomado de (ISO/IEC, 2017)

- **Array**

Un Array es una estructura delimitada por corchetes, uno Izquierdo y otro derecho “[”, “]” y dentro de ellas cero o más valores separados por una coma “,”.

Figura 10: Array en JSON

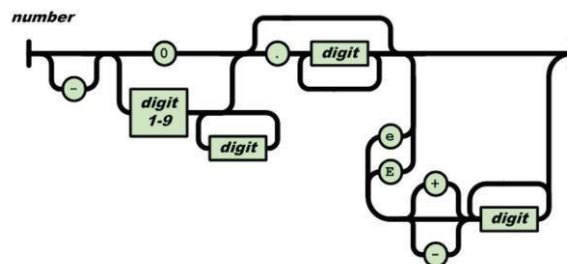


Nota: Tomado de (ISO/IEC, 2017)

- **Number**

Un number es una secuencia de dígitos decimales. Este puede incluir una parte fraccionaria prefijada por el punto “.”, un exponente prefijado por la letra “E”. Además, puede incluir un signo “+”, “-” pero es opcional.

Figura 11: Number en JSON



Nota: Tomado de (ISO/IEC, 2017)

- **String**

Un String es una serie de caracteres delimitadas por las comillas “”.

- **SOAP**

Según W3C (2000), SOAP es un modelo de intercambio de mensajes, estos mensajes siguen una estructura bien definida, pero no incluye la semántica de la data, las aplicaciones que intercambian los mensajes son los responsables de definir y acordar la semántica de los mensajes.

Los mensajes SOAP en su estructura está basado en XML y tienen un elemento raíz llamado Envelope.

- **Envelope**

Es el elemento raíz del documento XML, a su vez contiene a los elementos hijo llamados Header y Body. Es obligatorio en todo documento SOAP. El nombre XML de este elemento es “Envelope” (W3C, 2000).

- **Header**

Es una forma estandarizada y centralizada de agregar o definir características del documento. Es opcional, pero si aparece deberá ser el primer hijo de Envelope. El nombre XML de este elemento es “Header” (W3C, 2000).

- **Body**

Este elemento deberá estar presente en todo documento SOAP y posicionado justo después del Header, si este está presente.

c) Middleware

IBM (2022c), propone que Middleware es un producto de software que permite comunicación entre aplicaciones en una red distribuida, tanto entre sistemas internos a una empresa como para integrar con externos.

También considera que hay muchos tipos de middleware entre los que están el BUS SOA.

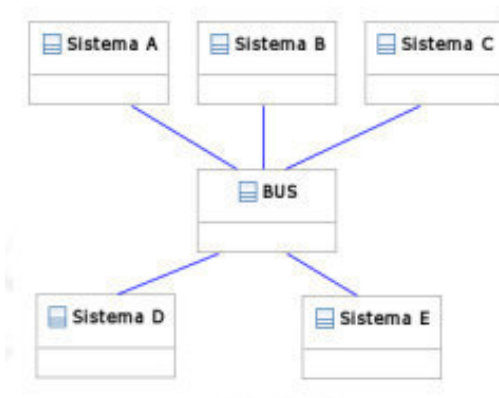
3.2.4.2 Trabajos relacionados:

a) **“Integración De Sistemas Orientada A Servicios”** (Jorge Luis Cabrera Chiappe, 2010)

En este trabajo autor muestra que al usar una arquitectura basada en Middleware se logra:

- Mejorar la eficiencia en el desarrollo y mantenimiento de interfaces, al tenerlas bien definidas, documentadas, bajo gobierno y custodia.
- Reducir el esfuerzo en el proceso de desarrollo de interfaces de integración al tomar el Middleware parte de las responsabilidades del proceso de integración como son:
 - Exponer protocolos diversos de comunicación.
 - Manejar de forma nativa la transferencia de mensajes.
 - Gestionar la transformación de formatos de mensajes.
 - Ayudar en la aplicación de patrones y políticas de seguridad en la integración entre sistemas.
- Mejorar la interoperabilidad entre las distintas plataformas dentro de la empresa y los externos, a través de la estandarización de los protocolos y formatos de mensaje.
- Aumentar la reutilización de interfaces y componentes desarrollados, esto se permite gracias a la definición formal de las interfaces, los contratos y la calidad de servicio comprometida en cada interfaz/Servicio.

Figura 12: Bus de integración



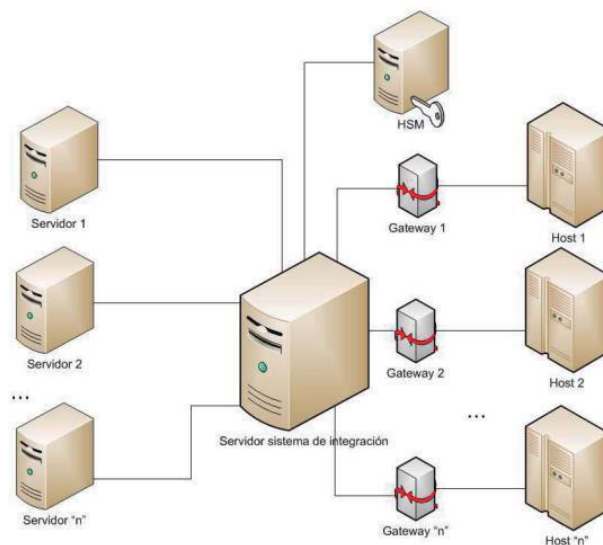
Nota: Extraído de (Cabrera, 2010)

Con esta arquitectura centrada en un BUS de servicios (Middleware) se logró integrar diferentes sistemas de forma eficiente, a bajo costo y en poco tiempo.

b) “Análisis Y Diseño de un Sistema Distribuido De Pago Middleware Orientado A La Mensajería Entre Una Entidad Bancaria Y Una Empresa Farmacéutica” (Tino Fabricio Reyna Colona, 2017)

En este trabajo el autor muestra cómo, al usar un middleware, interfaces genéricas y un esquema de mensajería bien definidos y estandarizados permite integrar entidades con giros de negocio muy diferentes, además de plataformas heterogéneas a través de mensajería y protocolos estandarizados.

Figura 13: Arquitectura física del sistema



Nota: extraído de Reyna Fabricio (2017)

La arquitectura adoptada en este trabajo muestra un BUS(Middleware) con rol central y protagonista, haciendo de interprete entre los interlocutores, permitiendo la integración eficiente, disminuyendo los riesgos y complicaciones de una integración entre sistemas heterogéneos.

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

De acuerdo con lo detallado en la sección 3.2.3. “Etapas y metodología”, el equipo trabajó con la metodología SCRUM, a continuación, se encuentra más detalle:

a) Equipo SCRUM

El equipo estaba formado por los siguientes roles, la autora del presente trabajo se desarrolló como líder técnico:

- Product Owner (Propietario del producto)
- Scrum Máster
- Team (Equipo): Arquitecto de Integración, el líder técnico, 2 desarrolladores y 1 analista de calidad

b) Épicas

En la tabla 4, se puede identificar las épicas que se trabajaron en el presente proyecto:

Tabla 4: Lista de épicas

Épicas	Descripción
EP01	Consulta de deuda en línea
EP02	Pago de deuda en línea
EP03	Extorno de deuda en línea

Nota. Fuente: Documentación funcional del proyecto

c) **Product Backlog**

El proyecto fue implementado en 6 sprints, y cada uno duró 2 semanas, en la tabla 5 se detalla la división de épicas en historias de negocio y en que sprint se implementó.

Tabla 5: Product Backlog

Épica	Historias de Negocio	SPRINT
Consulta de deuda en línea	Como canal recaudador deseo consumir los servicios de consulta genérica para validación parcial de las empresas recaudadoras	1
	Como canal recaudador deseo consumir los servicios de consulta genérica para validación completa de las empresas recaudadoras	2
Pago de deuda en línea	Como canal recaudador deseo consumir los servicios de pago genérica para validación parcial de las empresas recaudadoras	3
	Como canal recaudador deseo consumir los servicios de pago genérica para validación completa de las empresas recaudadoras	4
Extorno de deuda en línea	Como canal recaudador deseo consumir los servicios de extorno genérica para validación parcial de las empresas recaudadoras	5
	Como canal recaudador deseo consumir los servicios de extorno genérica para validación completa de las empresas recaudadoras	6

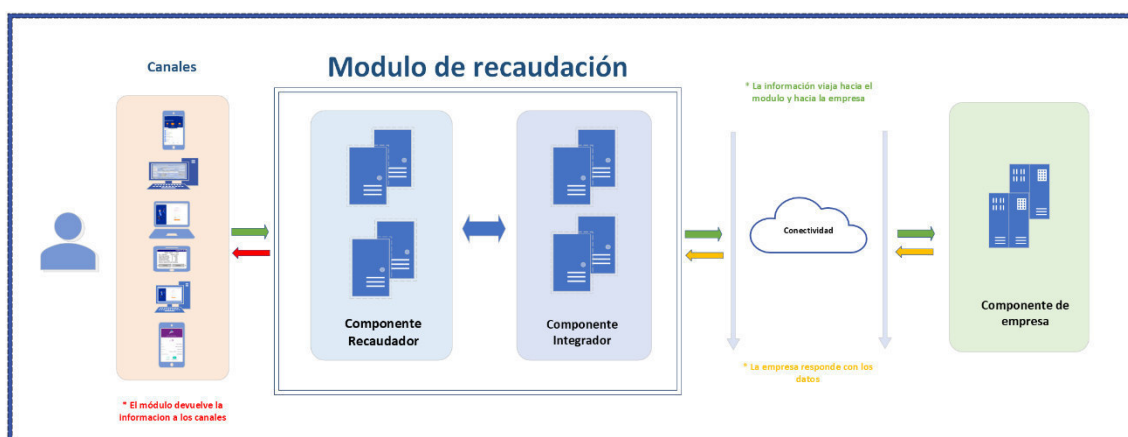
Nota. Fuente: Documentación funcional del proyecto

3.2.5.1 Fase 1 – Análisis y diseño de la integración

En esta fase empezamos con el requerimiento priorizado por el negocio, el cual es “implementar el pago de servicios en línea a través de una solución genérica”. Es así como se genera un primer diagrama funcional, esto se muestra en la figura 12, la comunicación inicia desde que un cliente busca y selecciona el servicio de la empresa

recaudadora en cualquiera de los canales que ofrece la entidad financiera y procede a identificarse con el código que le haya indicado la empresa recaudadora. Desde estos canales, la información se recibe y se envía al módulo de recaudación para procesar la información, donde se realizan validaciones iniciales. El componente integrador se encarga de enviar la información a la empresa recaudadora y procesar su respuesta, esta información es enviada a los canales para que sea mostrada al cliente. Luego que se ha realizado la consulta, el cliente seleccionará la deuda a pagar y esa información viajará nuevamente al módulo de recaudación para enviarlo a la empresa recaudadora a través del componente integrador y así obtener su respuesta. En caso la respuesta sea exitosa o de error, está será mostrada al cliente

Figura 14: Diagrama de flujo funcional de Recaudación

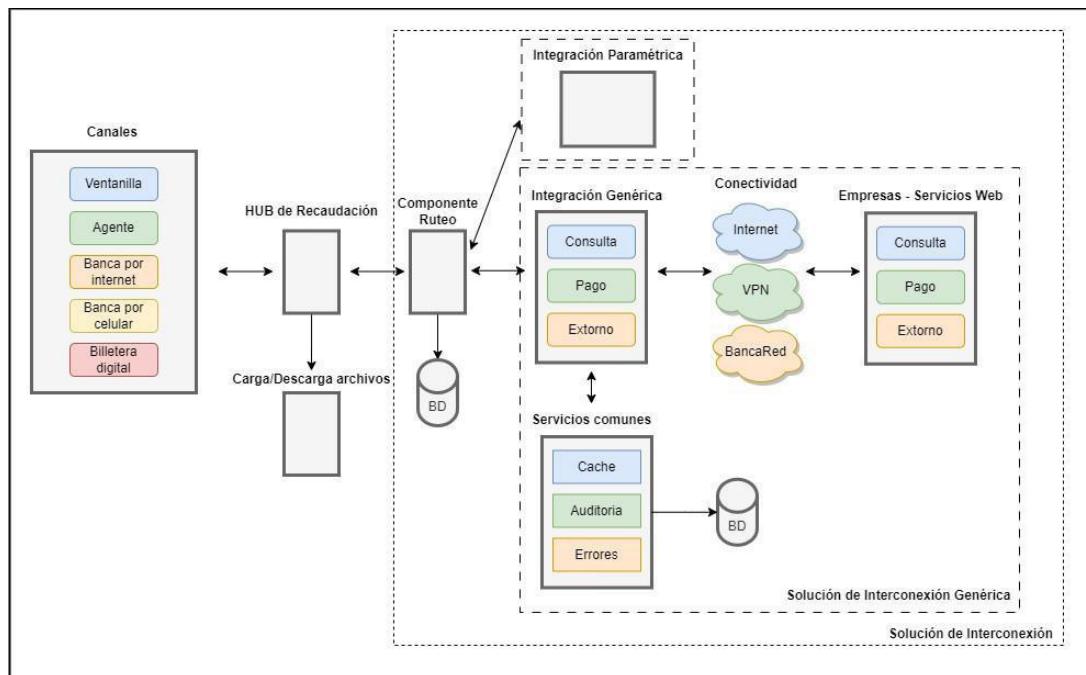


Nota: Fuente: Documentación funcional del proyecto

Con el diagrama funcional validado, el arquitecto de integración junto al líder técnico elaboró un diagrama de integración de los componentes necesarios que participaran en el flujo de recaudación, como se puede observar en la figura 13, la comunicación inicia desde cualquier canal con los que cuenta la entidad financiera, esto llega al hub de recaudación, aquí se encuentran configuradas las cuentas de las empresas y cuentan con un flag que indica si la interconexión es por transferencia de archivos o en línea. Cuando la empresa tiene activo el flag de interconexión en línea, el hub de recaudación envía la trama al componente de ruteo, este se encarga de validar en la base de datos los parámetros configurados por empresa, aquí se identifica si la empresa cuenta con una interconexión paramétrica, es decir, que se ha realizado un desarrollo específico para la empresa, o que irá por el flujo de una interconexión genérica, aquí se ha

identificado que es necesario un componente que se encargue de recibir la trama y, de acuerdo a la información que llega, identificar si se trata de una consulta, pago o extorno, y el tipo de validación, que puede ser parcial o completa. Este componente de integración genérica realizará la invocación de los servicios web expuestos por las empresas recaudadoras, a través de diferentes tipos de conectividad, los que pueden ser internet o VPN.

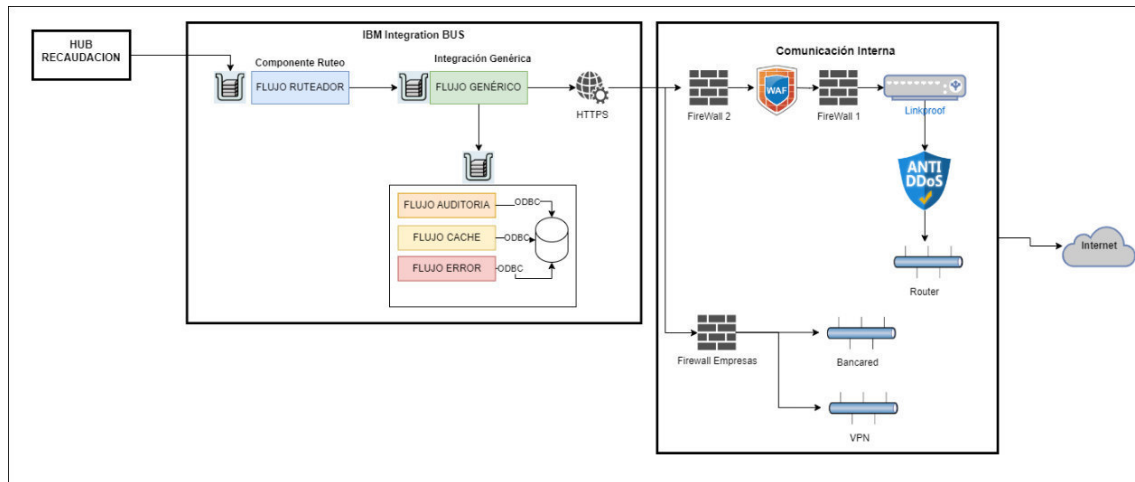
Figura 15: Diagrama de componentes



Nota: Fuente: Documentación funcional del proyecto

En la figura 16, se puede visualizar el diagrama de arquitectura de la integración: el Flujo Ruteador validará las peticiones de trama paramétrica o genérica, en el caso de las peticiones genéricas se invocará mediante una cola MQ al Flujo Genérico; el flujo genérico mediante una invocación HTTPS invocará a los servicios de las empresas recaudadoras y de acuerdo con el tipo de conectividad la invocación irá por los diferentes Firewall de la entidad financiera.

Figura 16: Diagrama de Arquitectura de Integración



Nota: Fuente: Documentación funcional del proyecto

Elaboración del diseño de los mensajes

Teniendo en cuenta los escenarios de integración que se deben cubrir, se deben implementar 6 estructuras de mensajes, la de consulta, pago y extorno de validación parcial, consulta, pago y extorno de validación completa. En la tabla 6 se procede a detallar la estructura del mensaje de solicitud que se diseñó para la consulta.

Tabla 6: Detalle Trama Solicitud Consulta Parcial

Campo	Descripción
rqUUID	Identificador único universal generado en la entidad financiera, para reconocer cada uno de los mensajes que forman parte de la transacción en curso.
operationDate	Fecha de la operación de la entidad financiera al momento de enviar la transacción.
operationNumber	Número de operación con el que se registra la transacción.
financialEntity	Código de la entidad financiera para identificarla.

channel	Código de canal que identifica el canal de la entidad financiera desde el cual se envió la transacción.
serviceId	Código de servicio que identifica al servicio involucrado en la transacción.
customerId	Código de cliente usado para identificar una deuda

Nota. Fuente: Elaboración propia

En la tabla 7, se muestra la estructura del mensaje de respuesta de la consulta parcial, las demás estructuras se detallarán en el Anexo 1:

Tabla 7: Detalle Trama Respuesta Consulta Parcial

Campo	Descripción
rqUUID	Identificador único universal que identifica el mensaje de solicitud con el que se inició la transacción.
resultCode	Código de respuesta de la transacción de consulta.
resultDescription	Descripción de código respuesta retornado.
operationDate	Fecha de proceso en la empresa en el momento en el que procesó la transacción solicitada.
operationNumberCompany	Nro. de operación generado por la empresa
customerName	Nombre del cliente asociado al identificador del pago registrado en los sistemas de la empresa recaudadora
merchantId	Código que identifica al comercio al cual se realiza la recaudación

paymentDetail	Descripción del producto que el cliente está adquiriendo
documentReference	Código retornado por la empresa recaudadora como referencia adicional al documento de pago.

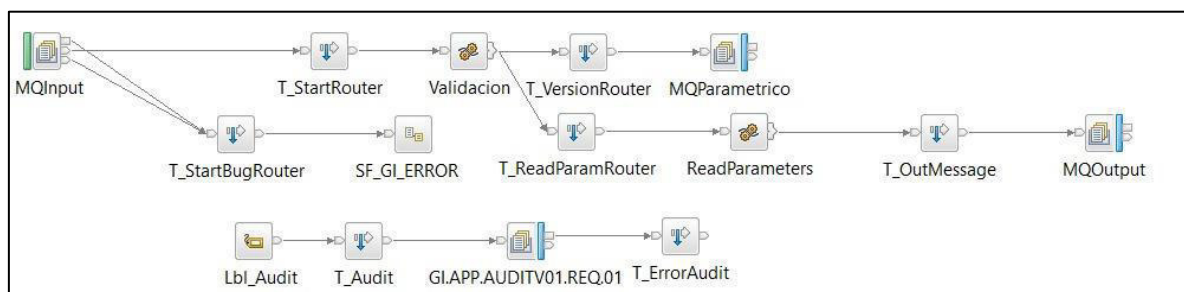
Nota. Fuente: Elaboración propia

3.2.5.2 Fase 2 – Construcción

Con el diseño de la arquitectura aprobado en la fase anterior y el diseño de los mensajes listos, se procede a realizar el desarrollo de los flujos:

a) **Flujo Ruteador:** En la figura 17, se puede observar el primer flujo que tiene como nombre “Ruteador”, este flujo se encarga de recibir las tramas provenientes del hub de recaudación, a través del nodo MQInput, este nodo se encarga de interpretar el mensaje de entrada y propagarlo al siguiente nodo: “Validación”, éste se encarga de validar si es un flujo paramétrico o genérico y según la configuración de la empresa (nodo “ReadParameters”), se envía a la cola correspondiente. Para los casos de las empresas genéricas, se arma la estructura necesaria y se envía a la cola genérica. Este flujo ruteador también se encarga de armar el mensaje para el registro de auditoría, y en caso de error, el mensaje se propaga a un subflujo de manejo de errores.

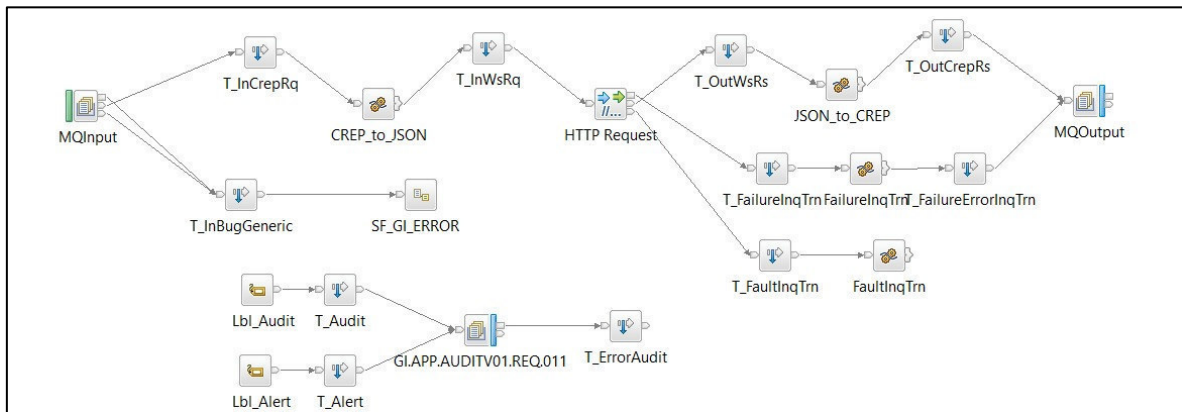
Figura 17: Flujo de integración RUTEADOR



Nota: Fuente: Documentación del proyecto

b) Flujo Genérico: En la figura 18, se puede observar el flujo que procesa la integración genérica, el nodo “MQInput” lee los mensajes provenientes del flujo ruteador y los propaga al siguiente nodo, este se encarga de validar el mensaje e identificar la operación, que puede ser: consulta, pago o extorno y también identifica el tipo de validación, que puede ser: parcial o completa, de acuerdo a esta información, se arma la estructura del mensaje para invocar al servicio de las empresas recaudadoras. También se encarga de armar el mensaje para el registro de auditoría. En caso de error se propaga a un subflujo de manejo de errores.

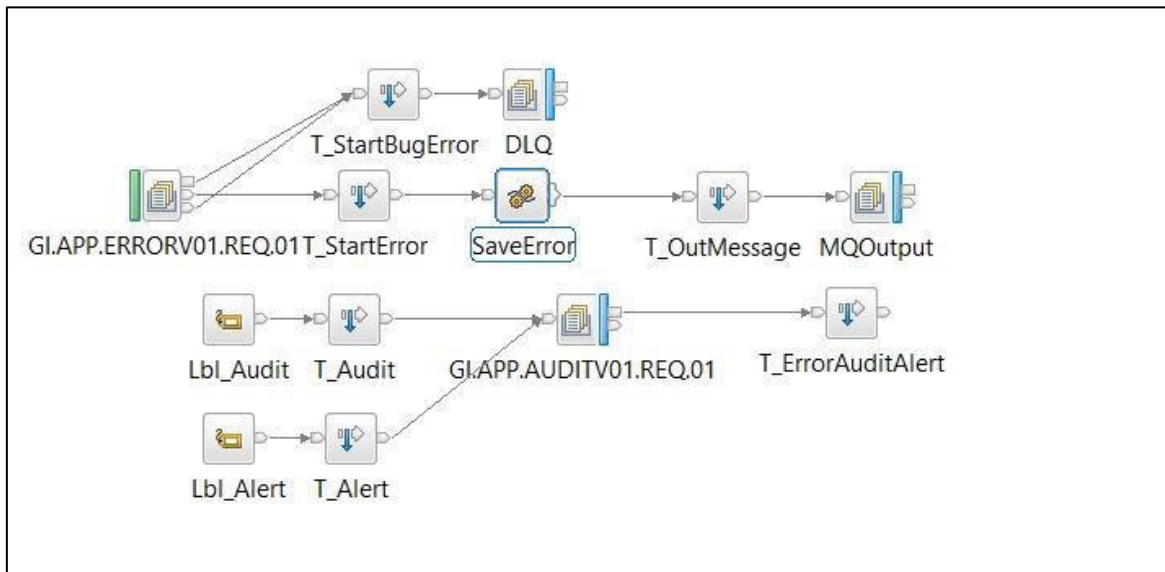
Figura 18: Flujo de integración GENERICO



Nota: Fuente: Documentación del proyecto

c) Flujo Error: En la figura 19, se puede observar el flujo de error, este se encarga de obtener el detalle del error, armar la estructura del mensaje de error para enviar el mensaje al hub de recaudación, también se encarga de generar el mensaje para auditoria.

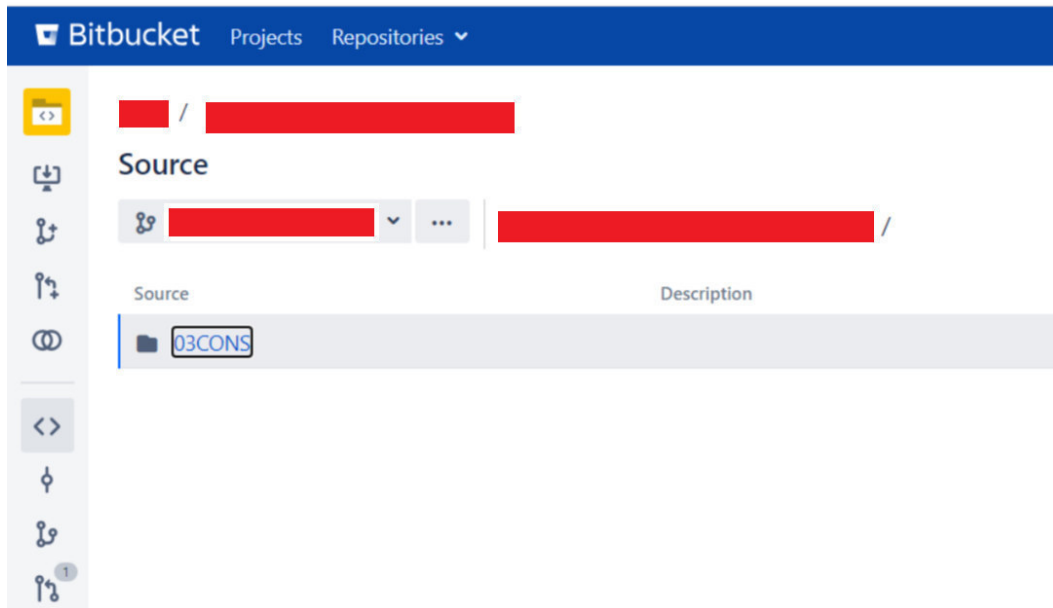
Figura 19: Flujo manejo ERROR



Nota: Fuente: Documentación del proyecto

Al finalizar el desarrollo, se iniciaron las pruebas unitarias y revisión de pares. Se realizó la documentación de todos los flujos desarrollados, esta información es guardada en el repositorio de la entidad financiera, luego se procedió a realizar el análisis de calidad y seguridad del código, este se guardó en el repositorio del proyecto en el Bitbucket siguiendo la estructura mostrada en la figura 20, luego el analista de calidad se encargó de elaborar los casos de prueba de integración. El líder técnico se encargó de generar la solicitud mediante un Ticket de cambio, esto se hace en la herramienta Jira, ahí se realizaron validaciones iniciales como la criticidad del cambio.

Figura 20: Repositorio Bitbucket usado para el código del proyecto

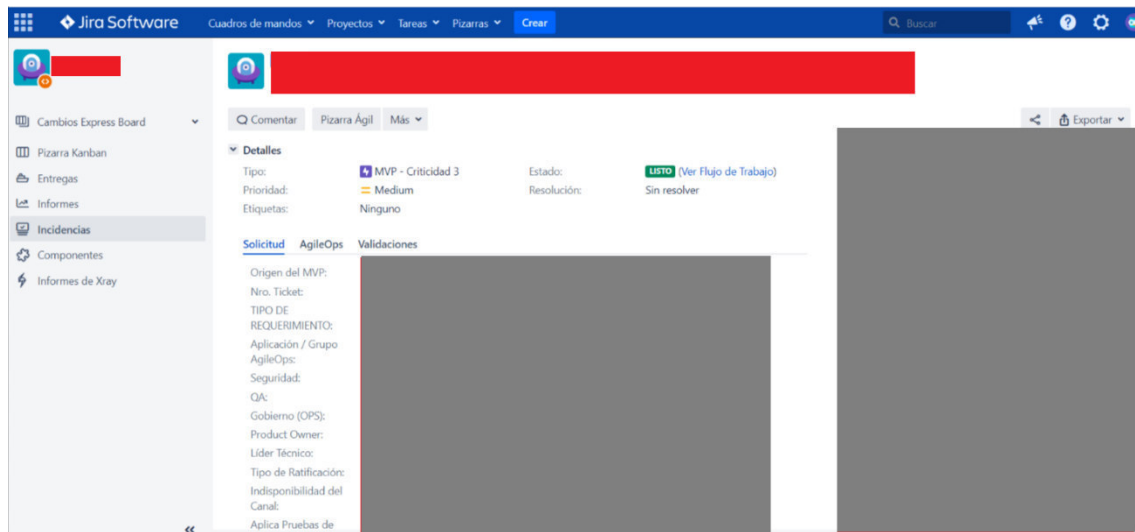


Nota: Fuente: Documentación del proyecto

3.2.5.3 Fase 3 – Congelamiento

En esta fase, todos los componentes de integración ya se encuentran guardados en el repositorio. El Ticket de cambio mostrado en la figura 21, creado en la fase anterior pasó por un flujo de aprobaciones, primero la del Arquitecto de integración y luego la del Analista de Seguridad. En esta fase, el equipo encargado de gestionar el pase se encargó de crear las tareas y derivarlas a los equipos especialistas que se encargaron de realizar la instalación de todos los componentes según las instrucciones que se encuentran detalladas en el ticket de cambio, en esta fase ya todos los componentes se encuentran instalados en el ambiente de certificación.

Figura 21: Ticket de Cambio para ejecutar el congelamiento



Nota: Fuente: Documentación del proyecto

3.2.5.4 Fase 4 – Pruebas

En esta fase el analista de calidad valida que todos los pasos ejecutados en la fase anterior hayan sido exitosos, esto nos confirma que el ambiente se encuentra listo para iniciar las pruebas. Al finalizar la primera etapa de validación, ya se ejecutaron todas las pruebas integrales, siguiendo los casos de prueba que previamente se habían definido. Se validan los resultados obtenidos y el analista de calidad se encarga de documentar el estado de todas estas pruebas. Como se muestra en la imagen 22, la solución de interconexión cuenta con una web administrativa que nos permite mostrar el resultado de las pruebas, aquí se visualizan todos los eventos registrados de auditoría. En caso el analista de calidad reporte un error en las pruebas, el ticket de cambio debe retornarse al estado inicial para que el líder técnico junto a los desarrolladores se encargue de corregir el error y volver a subir los cambios de las fuentes al repositorio e iniciar nuevamente con las conformidades iniciales del ticket de cambio.

Figura 22: Web Administrativa de Interconexión

216 elementos encontrados, mostrando 1 de 25.

MsgId	Secuencial	Fecha	Modulo	QMGR	Mensaje
41465120515043524550524845030316526158e233fc638	0001	17-11-2023 16.13.22	PREPARACION_CP_REQ	QPSOGORHEP0121	Ver
	0002	17-11-2023 16.13.22	SALIDA_WS_REQ	QPSOGORHEP0121	Ver
	0003	17-11-2023 16.13.24	SALIDA_WS_RES	QPSOGORHEP0121	Ver
	0004	17-11-2023 16.13.24	PREPARACION_CP_RES	QPSOGORHEP0121	Ver

Nota: Fuente: Documentación del proyecto

3.2.5.5 Fase 5 – Pase a producción

Con la conformidad de las pruebas en la fase anterior, se pide la aprobación del Product Owner y se procede a registrar el pase a producción en el calendario mostrado en la figura 23, aquí se reservan los recursos necesarios para ejecutar el pase a producción y también se colocan los datos de las personas que deben realizar la aprobación de esta reserva, estos son los jefes de los equipos de desarrollo y calidad.

Figura 23: Web Administrativa de Interconexión

Pases a Producción

RLM/CRQ/MVP Recurso Aplicación

Buscar

< > Hoy

Mes Semana Día

lun	mar	mié	jue	vie	sáb	dom
Bloqueo de Canales y 30	Bloqueo total 31	Bloqueo de Canales y 1	2	3	4	5
4 (INCT MVPBCP-169451) PO	4 (TCRQ CRQ000000290454) PO	4 (TCRQ CRQ000000290455) PO	4 (DWL CRQ000000291785) DEDICADO DWL TRIBU DATA	4 (TCRQ CRQ000000292058) PO	4 (MBBK MVPBCP-170642) DEDICADO MBK TRIBU CANALES DIG	4 (MBBK MVPBCP-170642) DEDICADO MBK TRIBU CANALES DIG
4 (TCRQ CRQ000000291554) PO	4 (SREC MVPBCP-165954) PO	4 (MGRL MVPLEGBCP-98	4 (RPER MVPBCP-170388	4 (TCRQ CRQ000000292059) PO	4 (MBBK MVPBCP-170642) DEDICADO MBK TRIBU CANALES DIG	4 (MBBK MVPBCP-170642) DEDICADO MBK TRIBU CANALES DIG
4 (VCEN CRQ000000291581) PO	4 (DPLZ MVPLEGBCP-980	4 (GRRC MVPBCP-16924	4 (WSPC OCD-4279) POOL	4 (TCRQ CRQ000000292059) PO	4 (MBBK MVPBCP-170642) DEDICADO MBK TRIBU CANALES DIG	4 (MBBK MVPBCP-170642) DEDICADO MBK TRIBU CANALES DIG
4 (TELE CRQ000000291409) DC	4 (DPLZ MVPLEGBCP-16270	4 (CRMC MVPLEGBCP-85	4 (LNSW CRQ000000291	4 (TELE CRQ000000291972) DC	4 (MBOB MVPBCP-171025) PO	4 (TCRQ CRQ000000292077) PO
+85 más	+9 más	+17 más	+185 más	+254 más	+100 más	+61 más
6	7	8	9	10	11	12
7 (DWL CRQ000000291785) DEDICADO DWL TRIBU DATA	7 (TCRQ CRQ000000293121) PO	7 (TCRQ CRQ000000293200) DERIVADO TCRQ IDT	7 (TCRQ CRQ000000293237) PO	7 (ALMB MVPLEGBCP-89372	7 (TCRQ CRQ000000293803) PO	7 (O365 CRQ000000293809) PO
21 (MBBK MVPBCP-170642) DEDICADO MBK TRIBU CANALES DIGITALES	21 (TCRQ CRQ000000293023) PO	21 (TCRQ CRQ000000293122) PO	21 (TCRQ CRQ000000293463) PO	21 (INCT MVPBCP-173530) PO	21 (TCRQ CRQ000000293603) PO	21 (TCRQ CRQ000000293603) PO
1 (WIDC CRQ000000292007) PO	1 (JAWA CRQ000000292608) DERIVADO JAVA COE CLOUD					

Nota: Fuente: Documentación del proyecto

Como se puede ver en la imagen 24, todos los equipos deben brindar su conformidad, lo siguiente es asistir al comité de pases, aquí se explicó el cambio y que componentes de la entidad financiera se verían afectados, luego de tener la aprobación de este comité, se gestionaron los recursos que ejecutarán el pase el día reservado en el

calendario. El día del pase, la persona encargada de gestionarlo creó las subtareas y las derivó con el equipo encargado de desplegar los componentes de integración, el analista de calidad realizó las pruebas para validar el pase y dar por atendido el pase con la solución finalmente instalada en el ambiente de producción.

Figura 24: Aplicación de aprobaciones



Aprobaciones			
Priori...	Título de la solicitud	Estado	Origen
	CONFORMIDAD OPS S	Aprobado	
	CONFORMIDAD DEV S	Aprobado	
	CONFORMIDAD QA SF	Aprobado	
	CONFORMIDAD LT SR	Aprobado	

Nota: Fuente: Documentación del proyecto

3.3 Evaluación

3.3.1 Evaluación Económica / Evaluación costo-beneficio

La evaluación económica se realizó de acuerdo con los costos del desarrollo del proyecto, mostrados en la tabla 8.

Tabla 8: Costo del desarrollo del proyecto

Rol	Costo/Mes (\$)	Tiempo (Mes)	Asignación (%)	Sub-Total (\$)
Product Owner	7,000.00	4	15	4,200.00
Arquitecto de Integración	7,000.00	4	30	8,400.00
Scrum Master	4,000.00	4	40	4,400.00
Líder técnico	4,500.00	4	90	16,200.00
Analista Programador 1	2,000.00	3	100	6,000.00
Analista Programador 2	2,000.00	3	100	6,000.00
Analista de Calidad	2,000.00	2	100	4,000.00
TOTAL				49,200.00

Nota. Fuente: Elaboración propia

El monto obtenido en la tabla anterior se sumó con los gastos para la implementación del proyecto tales como licencias, esto se puede observar en la tabla 9.

Tabla 9: Costos totales de inversión

Inversión	Monto (\$)
Costo del desarrollo del proyecto	49,200.00
Servidores y licencias	50,000.00
	99,200.00

Nota. Fuente: Elaboración propia

Con la implementación de la solución genérica y masiva, se ahorró el tiempo en afiliar a las empresas al sistema de recaudación de la entidad financiera. En la tabla 10 se presenta la comparación de ganancia antes y después de la implementación en un trimestre.

Tabla 10: Ganancia proyectada en 1 trimestre

DATOS	COMPARACION		RESULTADOS
	Antes de la implementación	Después de la implementación	Aumento de empresas afiliadas
Empresas afiliadas	2	12	10
Ganancias por las empresas afiliadas	\$7,500.00	\$45,000.00	\$37,500.00

Nota. Fuente: Elaboración propia

3.3.2 Interpretación del VAN y del TIR

La inversión del proyecto fue \$99,200.00 y generará un flujo de caja de \$150,000.00 por año. Se evaluó si el proyecto es viable con una tasa de interés del 12% en los dos primeros años, estos datos se presentan a continuación en la tabla 11:

Tabla 11: Resultado de la evaluación

Resultados del flujo de caja	
Inversión	99,200.00
Beneficio año 1	150,000.00
Beneficio año 2	150,000.00
Evaluación	
Tasa de interés	12%
VAN	61514.29
TIR	81.452

Nota. Fuente: Elaboración propia

El impacto del proyecto en la rentabilidad de la empresa fue muy destacable obteniendo un VAN de \$ 61514.29 y un TIR de 81.452%, demostrando la recuperación de la inversión en el primer año y la obtención de mayores ganancias en los años siguientes que utilizando esa inversión a una tasa de interés del 12%.

CAPITULO IV – REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

En el proyecto presentado en el presente trabajo, se requería integrar a la entidad financiera con las empresas recaudadoras utilizando un middleware, brindando una solución genérica y masiva, para esto se armó el equipo de desarrollo, el cual fue conformado por un Arquitecto de Integración, 1 Líder técnico, 2 desarrolladores y un analista de calidad. La autora del presente trabajo participó desde las etapas iniciales del proyecto hasta su puesta en producción, con el rol de Líder Técnico.

El proyecto significó un gran reto para mi persona, ya que era el primero en el que me desarrolle como líder de un equipo y más aún, sabiendo la importancia de este proyecto para la entidad financiera. Al realizar el presente proyecto tuve la oportunidad de conocer varias áreas y procesos de la entidad financiera, así mismo, tuve varios retos a lo largo del proyecto que se superaron, esto fue muy enriquecedor en mi experiencia profesional.

Pude conocer todos los escenarios de integración entre la entidad financiera y las empresas recaudadoras, también, cómo funciona el proceso de recaudación en general, los tipos de cuentas con las que trabaja la entidad financiera, la información que es relevante para el proceso y los términos que se manejan en este tipo de negocio. También conocí lo importante que es para las empresas recaudadoras lograr esta alianza con la entidad financiera, tanto para poder brindar mayores opciones de pagos a sus clientes, como para que sus procesos internos de conciliación sean mucho más sencillos.

Como parte de este proyecto, puedo asegurar que se cumplió con el objetivo de desarrollar una solución de integración en línea que sea genérica y masiva para el proceso de recaudación. Esto trajo muchos beneficios para la entidad financiera y las empresas recaudadoras que desean afiliarse a este servicio.

CAPITULO V – CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Se llegaron a las siguientes conclusiones:

- Se analizó el proceso funcional entre la entidad financiera y empresas recaudadoras (Sección 3.2.5.1 punto a), aquí se identificó los tipos de operaciones, tipos de validación y los tipos de conectividad que debían incluirse en la integración.
- Se analizó e identificó los componentes involucrados en la recaudación (Sección 3.2.5.1 punto b), esto se hizo para identificar el tipo de integración que se debía hacer con estos componentes.
- El Arquitecto de integración junto al líder técnico diseñaron los flujos de integración (Sección 3.2.5.1 punto c), se tomaron en cuenta todos los escenarios analizados anteriormente.
- El equipo de desarrollo junto al líder técnico se encargó de desarrollar y desplegar la solución en producción (Sección 3.2.5.1 punto d), realizando el seguimiento post pase a producción.
- Además, mediante el uso de middleware, se logró el objetivo de realizar la integración de la entidad financiera y las empresas recaudadoras (Sección 3.2.5.1 punto e), brindando una solución genérica y masiva para la recaudación en línea, cumpliendo con el objetivo de permitir las afiliaciones de nuevas empresas recaudadores en menor tiempo y costo.

5.2 Recomendaciones

A partir de la experiencia obtenida en el presente trabajo, se dan las siguientes recomendaciones:

- Mejorar el proceso de registro de la información que las empresas recaudadoras envían a la entidad financiera, esto mediante una web que les permita a las empresas recaudadoras registrar su información y pasar por una validación inicial.

- Mejorar el proceso de las pruebas de los servicios de las empresas recaudadoras, implementando pruebas automatizadas para las empresas que desean afiliarse a la interconexión genérica en línea.
- Dado los resultados positivos del presente proyecto, se recomienda implementar este modelo de integración en otros procesos de la entidad financiera.

5.3 Fuentes de Información

Amazon (2022) ¿Qué es la arquitectura orientada a servicios? Obtenido de amazon.com: <https://aws.amazon.com/es/what-is/service-oriented-architecture/#:~:text=you%20implement%20microservices%3F-,What%20is%20service%2Doriented%20architecture%3F,other%20across%20platforms%20and%20languages>

Cabrera Jorje (2010), Tesis: Integración de sistemas orientada a servicios

Eeles Peter y Cripps Peter (2010) Process of Software Architecting, The (English Edition).

ECMA (2017) ECMA-404 The JSON Data Interchange Syntax 2 nd Edition

Erl, Thomas (2005). Service-Oriented Architecture: Concepts, Technology, and Design (English Edition).

IBM (2019-2022), Advantages of XML Obtenido de ibm.com: <https://www.ibm.com/docs/i/7.5?topic=introduction-advantages-xml>

IBM (2022b), What is an ESB? Obtenido de ibm.com: <https://www.ibm.com/topics/esb>

IBM (2022c), What is Middleware? Obtenido de ibm.com: <https://www.ibm.com/topics/middleware>

IBM (2022a), What is SOA? Obtenido de ibm.com: <https://www.ibm.com/topics/soa>

ISO/IEC (2017) International Standard, Information technology — The JSON data interchange syntax 21778:2017(E)

Kruchten (2000). Kruchten, Philippe. The Rational Unified Process: An Introduction, 2nd ed.

Reyna Fabricio (2017), Tesis: Análisis Y Diseño De Un Sistema Distribuido De Pago Middleware Orientado A La Mensajería Entre Una Entidad Bancaria Y Una Empresa Farmacéutica

Len Bass, Clements Paul y Kazman Rick. (2013) Software Architecture in Practice (SEI Series in Software Engineering) (English Edition).

W3C (mayo 2000) Simple Object Access Protocol (SOAP) 1.1 Obtenido de w3.org: <https://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

5.4 Glosario

Core Bancario: es el sistema back-end de la entidad financiera, el cual procesa transacciones bancarias, contiene toda la información y configuración de las cuentas recaudadoras.

Cuenta recaudadora: Es una cuenta de ahorros que permite asociar el pago de servicios con un código de depositante.

Flag: es un marcador que se encuentra en el Core de recaudación, esto nos ayuda a activar o desactivar la recaudación en línea.

Validación completa: Es una modalidad de recaudación, aquí se valida tanto el código del cliente, como el monto a cancelar. Es necesario que la empresa envíe fechas diferentes de vencimiento para cada documento a pagar.

Validación parcial: Es una modalidad de recaudación, aquí se valida solo el código del cliente. Aquí no se envía fecha de vencimiento, porque el cliente no presenta deudas por pagar, sino que directamente ingresa un monto que desea abonar a un concepto de su servicio.

ANEXOS

ANEXO 1: DISEÑO DEL MENSAJE JSON

Se referencia en el punto 3.2.5.1. Fase 1 – Análisis y diseño de la integración, donde se especificó que se realizó el diseño de cada operación y validación:

Detalle Trama Solicitud Operación Pago Parcial

Campo	Descripción
rqUUID	Identificador único universal generado en la entidad financiera, para reconocer cada uno de los mensajes que forman parte de la transacción en curso.
operationDate	Fecha de la operación de la entidad financiera al momento de enviar la transacción.
operationNumber	Número de operación con el que se registra la transacción.
financialEntity	Código de la entidad financiera para identificarla.
channel	Código de canal que identifica el canal de la entidad financiera desde el cual se envió la transacción.
serviceId	Código de servicio que identifica al servicio involucrado en la transacción.
customerId	Código de cliente, es el mismo enviado en la consulta,
paymentType	Código de la forma en que se realizó el pago de la deuda.
amountTotal	Monto total de la transacción que se está pagando,
check	Contiene información del cheque con el cual se realiza el pago.
checkNumber	Es el número de cheque con el que el cliente realizó el pago de la deuda.
financialEntity	El código de la entidad financiera.
documentReference	Referencia adicional al documento del pago, este código es retornado por la empresa en la trama de consulta.

Detalle Trama Respuesta Operación Pago Parcial

Campo	Descripción
rqUUID	Identificador único universal que identifica el mensaje de solicitud con el que se inició la transacción.
resultCode	Código de respuesta de la transacción de pago.
resultDescription	Descripción de código respuesta retornado.
operationDate	Fecha de proceso en la empresa en el momento en el que procesó la transacción solicitada.
operationNumberCompany	Nro. de operación generado por la empresa
endorsement	Texto de refrendo que se muestra en los voucher de los canales cuando se procese una operación de pago exitosa.

Detalle Trama Solicitud Operación Externo Parcial

Campo	Descripción
rqUUID	Identificador único universal generado en la entidad financiera, para reconocer cada uno de los mensajes que forman parte de la transacción en curso.
operationDate	Fecha de la operación de la entidad financiera al momento de enviar la transacción.
operationNumber	Número de operación con el que se registra la transacción.
operationNumberAnnulment	Número de operación de pago registrado que se quiere exotrnar.
financialEntity	Código de la entidad financiera para identificarla.
customerId	Código de cliente, es el mismo enviado en el pago.
channel	Código que identifica al canal desde el cual se envió la transacción.

Detalle Trama Respuesta Operación Externo Parcial

Campo	Descripción
rqUUID	Identificador único universal que identifica el mensaje de solicitud con el que se inició la transacción.
resultCode	Código de respuesta de la transacción de extorno.
resultDescription	Descripción de código respuesta retornado.
operationDate	Fecha de proceso en la empresa en el momento en el que procesó la transacción solicitada.

Detalle Trama Solicitud Operación Consulta Completa

Campo	Descripción
rqUUID	Identificador único universal generado en la entidad financiera, para reconocer cada uno de los mensajes que forman parte de la transacción en curso.
operationDate	Fecha de la operación de la entidad financiera al momento de enviar la transacción.
operationNumber	Número de operación con el que se registra la transacción.
financialEntity	Código de la entidad financiera para identificarla.
channel	Código de canal que identifica el canal de la entidad financiera desde el cual se envió la transacción.
serviceId	Código de servicio que identifica al servicio involucrado en la transacción.

customerId	Código de cliente usado para identificar una deuda
------------	--

Detalle Trama Respuesta Operación Consulta Completa

Campo	Descripción
rqUUID	Identificador único universal que identifica el mensaje de solicitud con el que se inició la transacción.
resultCode	Código de respuesta de la transacción de consulta.
resultDescription	Descripción de código respuesta retornado.
operationDate	Fecha de proceso en la empresa en el momento en el que procesó la transacción solicitada.
operationNumberCompany	Nro. de operación generado por la empresa
customerName	Nombre del cliente asociado al identificador del pago registrado en los sistemas de la empresa recaudadora
merchantId	Código que identifica al comercio al cual se realiza la recaudación
documents	Grupo de documentos de pago contenidos en la transacción.
documentId	Número que identifica al documento de pago, es un identificador establecido por la empresa.
expirationDate	Fecha de vencimiento del documento de pago, indica cuando vencerá la deuda.

documentReference	Código retornado por la empresa como referencia adicional al documento de pago que será enviado.
paymentDetail	Descripción del producto que el cliente está adquiriendo.
amounts	Grupo de importes pertenecientes a un documento de pago
amountType	Concepto que describe el tipo de importe a mostrar, es obligatorio por cada documento de pago.
amount	Valor del monto asociado al tipo de importe.