



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

Implementación de una arquitectura ESB mediante el integrador empresarial WSO2 para la transmisión de documentos de terminales portuarios a SUNAT

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniera de Sistemas

AUTOR

Vidal CUPE QUISPE

ASESOR

César Augusto ALCÁNTARA LOAYZA

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

Cupe, V. (2021). *Implementación de una arquitectura ESB mediante el integrador empresarial WSO2 para la transmisión de documentos de terminales portuarios a SUNAT*. [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	Vidal Cupe Quispe
DNI	42575888
URL de ORCID	https://orcid.org/0000-0001-8738-8535
Datos de asesor	
Nombres y apellidos	Cesar Augusto Alcántara Loayza
DNI	09132297
URL de ORCID	https://orcid.org/0000-0003-3435-4555
Datos de investigación	
Línea de investigación	Desarrollo de modelos y aplicación de las tecnologías de información y comunicaciones.
Grupo de investigación	GESTIÓN, INVESTIGACIÓN, DESARROLLO E INNOVACIÓN EN LA DOCENCIA UNIVERSITARIA TIC
Agencia de financiamiento	No
Ubicación geográfica de la investigación	País: Perú Departamento: Lima Provincia: Lima Distrito: Surquillo Av. República de Panamá 4174 Piso 4 Latitud: -12.106289 Longitud: -77.018737
Año o rango de años en que se realizó la investigación	2021
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 20.00 horas del día 29 de junio del año 2021, se reunieron virtualmente los docentes designados como Miembros del Jurado del Trabajo de Suficiencia Profesional, presidido por el Mg. Luza Montero Cesar (Presidente), Mg. Cortez Vásquez Augusto Parcemón (Miembro) y el Mg. Alcántara Loayza Cesar Augusto (Miembro Asesor), usando la plataforma Meet (<https://meet.google.com/zce-gscs-mhs>), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: **“IMPLEMENTACIÓN DE UNA ARQUITECTURA ESB MEDIANTE EL INTEGRADOR EMPRESARIAL WSO2 PARA LA TRANSMISIÓN DE DOCUMENTOS DE TERMINALES PORTUARIOS A SUNAT”**, por el Bachiller **Cupe Quispe Vidal**; 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 habiendo efectuado la calificación correspondiente por los miembros del Jurado, el Bachiller obtuvo la nota de **18 DIECIOCHO**.

A continuación, el presidente de Jurados, el Mg. Luza Montero Cesar, declara al Bachiller **Ingeniero de Sistemas**.

Siendo las 21.00 horas, se levantó la sesión.

Presidente
Mg.. Luza Montero Cesar

Miembro
Mg. Cortez Vásquez Augusto Parcemon

Miembro Asesor
Mg. Alcántara Loayza Cesar Augusto

FICHA CATALOGRÁFICA

IMPLEMENTACIÓN DE UNA ARQUITECTURA ESB MEDIANTE EL INTEGRADOR EMPRESARIAL WSO2 PARA LA TRANSMISIÓN DE DOCUMENTOS DE TERMINALES PORTUARIOS A SUNAT

CUPE QUISPE, VIDAL

LIMA – PERU, 2021

ASESOR: CESAR ALCÁNTARA LOAYZA

INGENIERÍAS / TECNOLOGÍA DE INFORMACIÓN Y COMUNICACIÓN / INGENIERÍA DE SOFTWARE

TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS – FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA – ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS

Formato 28 x 20 cm

Páginas: x, 81

DEDICATORIA

A mis padres

Les dedico este trabajo

Por su gran apoyo durante todos mis estudios

Sin ellos hubiera sido más difícil llegar a ser un gran profesional

A mi hermana

Por estar siempre a mi lado apoyándome en todo momento.

AGRADECIMIENTOS

A mis familiares y amigos por siempre compartir conmigo momentos inolvidables.

A la Facultad de Ingeniería de Sistemas e Informática de la UNMSM por brindarme los mejores años en aprendizaje junto a los mejores profesores y compañeros.

A las empresas donde he podido trabajar que me permitieron aprender y asumir nuevos retos tecnológicos.

A mi asesor Cesar Alcántara Loayza por la excelente orientación y guía que me ayudaron a terminar este trabajo de forma satisfactoria.

TABLA DE CONTENIDOS

FICHA CATALOGRÁFICA.....	ii
DEDICATORIA.....	iii
AGRADECIMIENTOS.....	iv
RESUMEN	xi
ABSTRACT	xii
INTRODUCCION	13
CAPITULO I - TRAYECTORIA PROFESIONAL.....	15
CAPITULO II - CONTEXTO EN EL QUE SE DESAROLLO LA EXPERIENCIA.....	18
2.1 EMPRESA - ACTIVIDAD QUE REALIZA	18
2.2 VISION	20
2.3 MISION.....	20
2.4 ORGANIZACION DE LA EMPRESA.....	21
2.5 AREA, CARGO Y FUNCIONES DESEMPEÑADAS	21
2.6 EXPERIENCIA PROFESIONAL REALIZADA EN LA ORGANIZACIÓN.....	22
CAPITULO III - ACTIVIDADES DESARROLLADAS	23
3.1 SITUACION PROBLEMÁTICA.....	23
3.1.1 DEFINICION DEL PROBLEMA	23
3.2 SOLUCION	23
3.2.1 OBJETIVO GENERAL.....	23
3.2.2 OBJETIVOS ESPECÍFICOS.....	23
3.2.3 ALCANCE.....	24
3.2.4 ETAPAS Y METODOLOGIA	25
3.2.5 FUNDAMENTOS UTILIZADOS.....	132
3.2.6 IMPLEMENTACIÓN DE LAS AREAS, PROCESOS, SISTEMAS Y SUS BUENAS PRÁCTICAS	141
3.3 EVALUACIÓN.....	142
3.3.1 EVALUACIÓN ECONÓMICA.....	142
3.3.2 INTERPRETACIÓN DEL VAN Y DEL TIR.....	143
CAPITULO IV. REFLEXION CRITICA DE LA EXPERIENCIA.....	144
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES.....	146
5.1 CONCLUSIONES.....	146
5.2 RECOMENDACIONES	146
FUENTES DE INFORMACIÓN.....	147

GLOSARIO	148
ANEXOS	149
1. ANEXO 01 – Certificado de trabajo en Arquitectura Cloud.....	149
2. ANEXO 02 – Estándares de Base de Datos.....	151

INDICE DE FIGURAS

Figura 1-Designación de Technology Partner de AWS	18
Figura 2-Principales clientes de Complexless	20
Figura 3-Organigrama de Complexless	21
Figura 4-Alcance Organizacional en TPP	24
Figura 5-Equipo básico scrum del proyecto	25
Figura 6-Prototipo de pantalla de monitoreo IRM	28
Figura 7-Prototipo de pantalla de monitoreo RSA	28
Figura 8-Épicas e Historias de Usuario del Proyecto	29
Figura 9-Sprint Backlog del Sprint 01	29
Figura 10-Modelo de datos para Transmisiones	36
Figura 11-Sprint Backlog del Sprint 02	50
Figura 12-Diagrama de Solución Tecnológica	50
Figura 13-Diagrama de Flujo del componente Facade	53
Figura 14-Diagrama de Flujo del componente Transmisión	54
Figura 15-Diagrama de Flujo del componente Consulta	55
Figura 16-Diagrama de Flujo del componente Reintento de Transmisiones	56
Figura 17-Diagrama de Flujo del componente Reintento de Consulta	57
Figura 18-Diagrama de solución de escalabilidad de Reintento de Transmisiones	58
Figura 19-Diagrama de solución de escalabilidad de Reintento de Consulta	59
Figura 20-Estructura de proyecto del componente Facade	61
Figura 21-Estructura de proyecto del componente Transmisión	63
Figura 22-Estructura de proyecto del componente Consulta	65
Figura 23-Estructura de proyecto del componente Reintento Transmisión	67
Figura 24-Estructura de proyecto del componente Reintento Consulta	69
Figura 25-Sprint Backlog del Sprint 03	121
Figura 26-Cliente de petición de prueba de Caso 1	122
Figura 27-Configuración de métricas de prueba de Caso 1	123
Figura 28-Resultado de peticiones de prueba de Caso 1	123
Figura 29-Resumen estadístico de prueba de Caso 1	124
Figura 30-Cliente de petición de prueba de Caso 2	124
Figura 31-Configuración de métricas de prueba de Caso 2	125
Figura 32-Resultado de peticiones de prueba de Caso 2	125

Figura 33-Resumen estadístico de prueba de Caso 2	126
Figura 34-Diagrama de despliegue de componentes de arquitectura	128
Figura 35-Sprint Backlog del Sprint 04	128
Figura 36-Diagrama de componentes backend	129
Figura 37-Pantalla de cabecera de monitoreo de IRM	130
Figura 38-Pantalla de detalle de monitoreo de IRM	130
Figura 39-Pantalla de reporte de monitoreo de IRM	131
Figura 40-Pantalla de gráfico estadístico de monitoreo de IRM	131
Figura 41-Pantalla de cabecera de monitoreo de RSA	131
Figura 42-Pantalla de detalle de monitoreo de RSA	132
Figura 43-Pantalla de reporte de monitoreo de RSA	132
Figura 44-Sistema de Despacho Aduanero	133
Figura 45-Flujo de Scrum para un Sprint	136
Figura 46-Escalamiento vertical	138
Figura 47-Escalamiento horizontal	138
Figura 48-Diagrama de una arquitectura ESB	139

INDICE DE TABLAS

Tabla 1. Porcentaje de mercado en servicios de TI de los últimos 3 años.....	19
Tabla 2. Product Backlog del proyecto	27
Tabla 3. Plantilla de Ingreso y recepción de mercancías.....	30
Tabla 4. Campos de negocio para el Ingreso y recepción de mercancías	31
Tabla 5. Mapeo de campos de negocio con las etiquetas de la plantilla de IRM.....	32
Tabla 6. Plantilla de Registro de salida de mercancías.....	33
Tabla 7. Campos de negocio para el Registro de salida de mercancías.....	34
Tabla 8. Mapeo de campos de negocio con las etiquetas de la plantilla de RSA	35
Tabla 9. Scripts de generación de objetos de la Base de Datos de Transmisiones	37
Tabla 10. Código de entrada de evento FacadeInboundEP	70
Tabla 11. Código de flujo de proceso FacadeMainSeq.....	71
Tabla 12. Código de flujo de proceso FacadeMainErrorSeq	77
Tabla 13. Código de entrada de evento TransmisionInboundEP.....	78
Tabla 14. Código de flujo de proceso TransmisionMainSeq	79
Tabla 15. Código de flujo de proceso TransmisionMainErrorSeq.....	84
Tabla 16. Código de flujo de proceso TransmisionSunatIRMSeq	84
Tabla 17. Código de flujo de proceso TransmisionSunatIRMSerrorSeq	86
Tabla 18. Código de flujo de proceso TransmisionSunatRSASeq.....	86
Tabla 19. Código de flujo de proceso TransmisionSunatRSAErrorSeq	89
Tabla 20. Código de entrada de evento ConsultaInboundEP.....	91
Tabla 21. Código de flujo de proceso ConsultaMainSeq.....	92
Tabla 22. Código de flujo de proceso ConsultaMainErrorSeq.....	93
Tabla 23. Código de flujo de proceso ConsultaSunatSeq	93
Tabla 24. Código de flujo de proceso ConsultaSunatErrorSeq.....	97
Tabla 25. Código de entrada de evento ReintentoEnvioInboundEP.....	98
Tabla 26. Código de entrada de evento ReintentoEnvioDBEventInboundEP	99
Tabla 27. Código de flujo de proceso ReintentoEnvioMainSeq.....	100
Tabla 28. Código de flujo de proceso ReintentoEnvioMainErrorSeq	103
Tabla 29. Código de flujo de proceso ReintentoEnvioSunatIRMSeq	103
Tabla 30. Código de flujo de proceso ReintentoEnvioSunatIRMSerrorSeq.....	105
Tabla 31. Código de flujo de proceso ReintentoEnvioSunatRSASeq	107
Tabla 32. Código de flujo de proceso ReintentoEnvioSunatRSAErrorSeq	110
Tabla 33. Código de entrada de evento ReintentoConsultaInboundEP	112
Tabla 34. Código de entrada de evento ReintentoConsultaDBEventInboundEP	113
Tabla 35. Código de flujo de proceso ReintentoConsultaMainSeq	114
Tabla 36. Código de flujo de proceso ReintentoConsultaMainErrorSeq	117
Tabla 37. Código de flujo de proceso ReintentoConsultaSunatSeq.....	117
Tabla 38. Código de flujo de proceso ReintentoConsultaSunatErrorSeq	120
Tabla 39. Métricas de pruebas de estrés.....	121
Tabla 40. Casos de pruebas de estrés	121
Tabla 41. Normas aduaneras.....	141
Tabla 42. Costos del proyecto	142

Tabla 43. Cálculo del VAN	143
--	------------

RESUMEN

AUTOR: Cupe Quispe, Vidal

Las empresas portuarias peruanas diariamente necesitan cumplir el control aduanero de la Superintendencia Nacional de Administración Tributaria - SUNAT, para ello envían documentos vía electrónica de los ingresos y salidas de las mercancías que se mueven en sus almacenes ubicados en los puertos del Perú.

El presente trabajo de suficiencia profesional trata de la implementación de una arquitectura de integración en una empresa portuaria cuyo principal objetivo es mejorar la escalabilidad de la transmisión de documentos de ingresos y salidas a SUNAT. Para la implementación de la solución se usó la plataforma de integración empresarial wso2 que permite diseñar arquitecturas ESB en un entorno de servicios en nube.

Palabras claves: Arquitectura de integración, ESB, wso2, integrador empresarial, control aduanero, escalabilidad, SUNAT.

ABSTRACT

AUTHOR: Cupe Quispe, Vidal

The Peruvian port companies daily need to comply with the customs control of the National Superintendency of Tax Administration - SUNAT, for this they send documents electronically of the entries and exits of the goods that move in their warehouses located in the ports of Peru.

The present work of professional sufficiency is about the implementation of an integration architecture in a port company whose main objective is to improve the scalability of the transmission of incoming and outgoing documents to SUNAT. For the implementation of the solution, the wso2 business integration platform was used, which allows the design of ESB architectures in a cloud service environment.

Keywords: Integration architecture, ESB, wso2, business integrator, customs control, scalability, SUNAT.

INTRODUCCION

El presente trabajo profesional desarrolla la implementación de una arquitectura ESB mediante el integrador empresarial wso2 para la transmisión de documentos desde una empresa portuaria a SUNAT.

Para el cumplimiento del control aduanero las empresas portuarias necesitan transmitir vía electrónica las diferentes transacciones de las operaciones diarias de ingreso y salida de mercancías almacenadas en los contenedores de las naves, la transmisión de los manifiestos y operaciones asociadas, se envían en formato CUSCAR- XML, para su respectiva validación y aceptación por parte del Sistema Electrónico de Intercambio de Datos (SEIDA) de SUNAT.

Para enviar estas transacciones a SUNAT la empresa utilizó un sistema onpremise desktop legado, que permitía hacer los envíos solo desde la red interna impidiendo que nuevas plataformas tecnológicas de atención puedan integrarse. Otros de los problemas constantes era que durante la transmisión de documentos electrónicos los errores de comunicación hacían perder tiempo valioso debido a que se tenía que volver a enviar la transmisión por el operador en algunos casos superando las horas límites de control, lo cual repercutía en multas.

Por las razones antes mencionadas es que la empresa portuaria decidió tercerizar la implementación de una nueva arquitectura de integración con SUNAT desplegada en nube que permita que otras canales y plataformas tecnológicas puedan integrarse, así como también que esta nueva arquitectura tenga la robustez y la tolerancia a fallos para el cumplimiento a tiempo de los controles exigidos por SUNAT.

El trabajo se realizó desarrollando los siguientes capítulos:

En el Capítulo I, detallaré mi trayectoria profesional, la que refleja la experiencia adquirida y que permitió que pueda ejecutar de manera exitosa el proyecto de implementación de una arquitectura ESB mediante el integrador empresarial wso2 para la transmisión de documentos en una empresa portuaria a SUNAT.

En el Capítulo II, explicaré el contexto en el que se desarrolló la experiencia, describiendo a la empresa Complexless, su visión y misión, organigrama, áreas, cargo y funciones desempeñadas, así como la experiencia profesional realizada en la organización.

En el Capítulo III, detallaré el problema presentado, los objetivos y alcance, las etapas y metodología empleadas, los fundamentos utilizados e implementación.

En el Capítulo IV, realizaré una reflexión crítica de la experiencia.

En el Capítulo V, explicaré las conclusiones y recomendaciones del presente trabajo profesional.

CAPITULO I - TRAYECTORIA PROFESIONAL

PRESENTACION PROFESIONAL

Soy un profesional **BACHILLER** en **INGENIERIA DE SISTEMAS**, con amplia experiencia y especialidad en Arquitectura de Software usando diferentes tipos de tecnologías web, móviles, integración y cloud implementando sistemas en organismos públicos y empresas privadas de reconocido nivel nacional e internacional.

Mi experiencia profesional se da con especial énfasis en las siguientes especialidades:

- Arquitectura de Software,
- Consultoría en Cloud Computing,
- Análisis, diseño y construcción de soluciones tecnológicas,

Busco la mejora continua basados en el constante aprendizaje de nuevas tecnologías y en asumir el reto de implementar nuevos proyectos empresariales.

FORMACION ACADEMICA

Bachiller en Ingeniería de Sistemas

- Centro de Estudios: Universidad Nacional Mayor de San Marcos – Facultad de Ingeniería de Sistemas e Informática – Escuela Académico Profesional de Ingeniería de Sistemas.
- Periodo y Duración: 2003 – 2009

EXPERIENCIA PROFESIONAL

Octubre 2020 – Abril 2021	COMPLEXLESS S.A.C <ul style="list-style-type: none">• Cargo: Consultor en Arquitecto Cloud.• Funciones:<ul style="list-style-type: none">- Diseño y desarrollo de flujos de arquitectura de integración ESB con wso2.- Desarrollo de microservicios con Spring Boot, Docker y API Gateway- Diseño de componentes en AWS.• Logros:<ul style="list-style-type: none">- Implementación en el cliente “Terminales Portuarios Peruanos SAC” para la implementación de arquitectura de integración ESB usando wso2 para la transmisión de documentos a SUNAT.
Agosto 2019 - Septiembre 2020	TAX TECH S.A.C <ul style="list-style-type: none">• Cargo: Arquitecto de Software.

	<ul style="list-style-type: none"> • Funciones: <ul style="list-style-type: none"> - Diseño de Arquitectura Frontend y Backend. - Diseño de Arquitectura de integración con HBASE BigData e integración con DataPower. - Diseño de Componentes en Airflow con Python. • Logros: <ul style="list-style-type: none"> - Implementación en el cliente “Claro SA” del Sistema Fiscal que analiza, gestiona la información y documentación de sustento para la fiscalización de SUNAT.
Agosto 2017 - Diciembre 2018	PERÚ SOFTWARE FACTORY S.A.C <ul style="list-style-type: none"> • Cargo: Consultor en Arquitectura de Software. • Funciones: <ul style="list-style-type: none"> - Diseño de arquitectura web e integración en Oracle Weblogic 12c - Diseño de arquitectura móvil en Android. - Implementación de entornos de despliegue y testing en GIT, Bitbucket, Jenkins, Tomcat, Maven, Weblogic. • Logros: <ul style="list-style-type: none"> - Implementación del Sistema que permite registrar formularios relacionados a supervisión y cuestionarios a través de un aplicativo móvil.
Junio 2016 - Julio 2017	AVANCES TECNOLÓGICOS S.A.C <ul style="list-style-type: none"> • Cargo: Consultor de Software • Funciones: <ul style="list-style-type: none"> - Mantenimiento de aplicaciones híbridas en Android y iOS. • Logros: <ul style="list-style-type: none"> - Mantenimiento en el cliente “Unique Yanbal” de las aplicaciones móviles para realización de pedidos y administración de contactos.
Mayo 2013 - Julio 2016	ZOLUXIONES CONSULTING S.A.C <ul style="list-style-type: none"> • Cargo: Arquitecto de Software. • Funciones: <ul style="list-style-type: none"> - Diseño de la arquitectura para Android y iOS. - Diseño de Web Services en WAS e Integración con MQ Server • Logros: <ul style="list-style-type: none"> - Implementación en el “Banco Santander” del aplicativo móvil que consiste en realizar transacciones de pago de productos. - Implementación en “AFP Integra” del Sistema de atención de consultas y reclamos.

<p>Abril 2012 - Julio 2012</p>	<p>VATES S.A (Córdoba, Argentina)</p> <ul style="list-style-type: none"> • Cargo: Senior Software Engineer. • Funciones: <ul style="list-style-type: none"> - Análisis técnico. - Diseño de pruebas automatizadas • Logros: <ul style="list-style-type: none"> - Implementación de pruebas automatizadas en el sistema domótico que integra dispositivos dentro de un hogar para que sean controlados de forma remota.
<p>Febrero 2011 - Marzo 2012</p>	<p>VIRTUAL TELECOM S.A.C</p> <ul style="list-style-type: none"> • Cargo: Arquitecto de Software. • Funciones: <ul style="list-style-type: none"> - Análisis técnico y funcional. - Configuración Asterisk. - Diseño de la arquitectura de componentes. • Logros: <ul style="list-style-type: none"> - Implementación de sistema que ofrece servicios en la nube de una central telefónica como VoIP, distribución automática de llamadas, conferencias, buzón de voz bajo la integración de Asterisk y Java
<p>Agosto 2010 - Febrero 2011</p>	<p>NOVATRONIC S.A</p> <ul style="list-style-type: none"> • Cargo: Analista Programador Java. • Funciones: <ul style="list-style-type: none"> - Análisis y programación en Java y Oracle. - Configuración plataforma SIX. - Configuración Linux • Logros: <ul style="list-style-type: none"> - Implementación en el cliente Interbank de la Plataforma de servicios transaccionales que permite la suscripción de clientes a alertas y notificaciones según las operaciones en sus cuentas.

CAPITULO II - CONTEXTO EN EL QUE SE DESAROLLO LA EXPERIENCIA

2.1 EMPRESA - ACTIVIDAD QUE REALIZA

Complexless es una empresa peruana fundada el 2011 como un startup, brinda servicios de consultoría, outsourcing e implementación de soluciones en nube a través de sus unidades de negocios de Aplicaciones Web, Aplicaciones Móviles, Factura electrónica, Cloud Services.

Aplicaciones Web.

La unidad maqueta, diseña y desarrollo aplicaciones web usando tecnología backend y frontend aplicando una arquitectura de microservicios.

Aplicaciones Móviles.

La unidad desarrolla aplicaciones móviles nativas e híbridas en todas las plataformas (iOS, Android, Windows Phone) para Smartphones y Tablets.

Factura electrónica.

En la unidad se implementó el producto FACTUS con las mejores prácticas para un sistema altamente transaccional. A la fecha dicho sistema procesa 8 millones de documentos por mes y atiende a más de 800 clientes.

Cloud Services.

La unidad cuenta con los siguientes servicios especializados de servicios en nube a disposición de nuestros clientes: Cloud Architecture, managed Services y managed hosting.

Actualmente la empresa cuenta con la designación de ser “Technology Partner” de AWS.



Figura 1- Designación de Technology Partner de AWS (Fuente: LinkedIn Complexless)

El crecimiento de la empresa ha sido constante, principalmente en los servicios a cloud services.

Tabla 1. Porcentaje de mercado en servicios de TI de los últimos 3 años

SERVICIOS TI	2019	2020	2021
Aplicaciones Web	2%	2%	2%
Aplicaciones móviles	3%	3%	4%
Factura electrónica	2%	3%	3%
Cloud Services	3%	4%	6%

Datos de la Empresa:

RUC: 20543314529

Razón Social: COMPLEXLESS S.A.C

Tipo Empresa: SOCIEDAD ANÓNIMA CERRADA

Condición: Activo

Fecha Inicio Actividades: 01/Junio/2011

Actividad Comercial: Otras actividades de tecnología de la información

CIU: 72909

Dirección Legal: Av. Andrés Aramburú Nro. 856 Dpto. 302

Distrito/Ciudad: Surquillo

Departamento: Lima, Perú

Principales clientes:

Privadas: BCP, Scotiabank, PRIMAX, Banco Falabella, TPP

Públicas: Banco de la Nación, SUNAT, CONCYTEC, VUCE Perú



Figura 2- Principales clientes de Complexless (Fuente: Elaboración Propia)

2.2 VISION

Ser una empresa líder en el mercado regional usando tecnología Cloud para la implementación de proyectos innovadores.

2.3 MISION

Implementar soluciones de negocio basados en tecnologías especializadas con pensamiento innovador.

2.4 ORGANIZACION DE LA EMPRESA

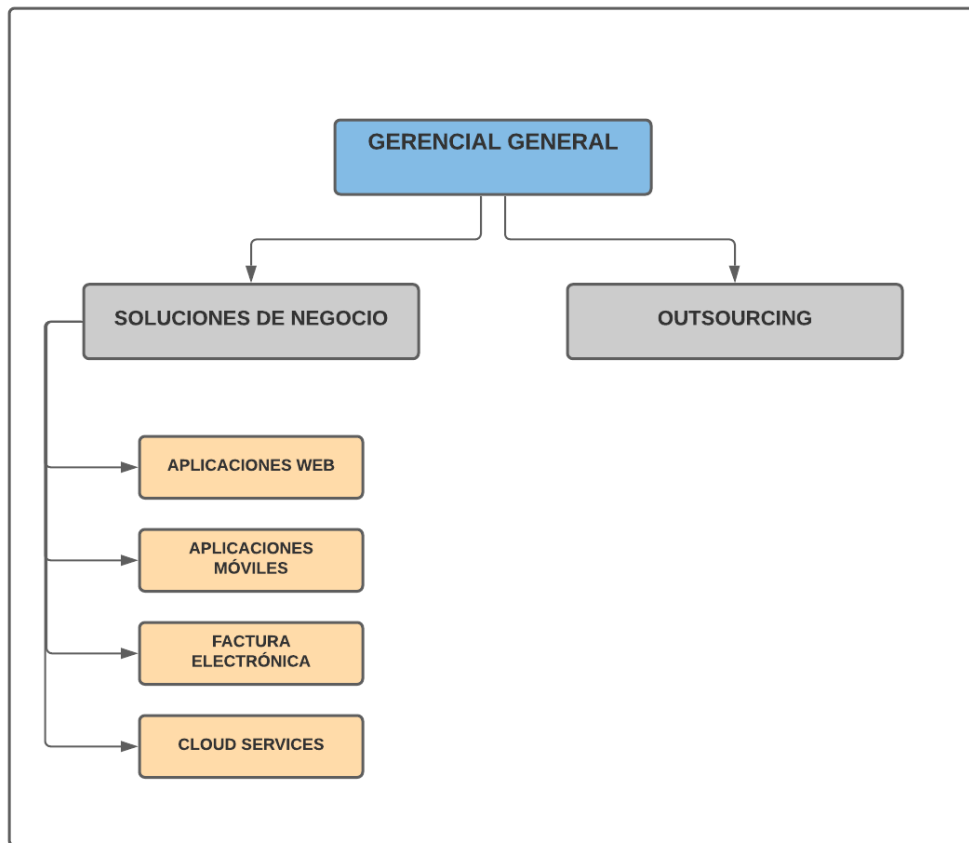


Figura 3-Organigrama de Complexless (Fuente: Elaboración Propia)

2.5 AREA, CARGO Y FUNCIONES DESEMPEÑADAS

Desempeñé el cargo de **Consultor en Arquitecto Cloud** en el área de Cloud Services para el cliente Terminales Portuarios Peruanos (TPP), desde el 26 de Octubre del 2020 hasta 30 de Abril del 2021.

Mis funciones fueron las siguientes:

- a) Diseño y desarrollo de flujos de arquitectura de integración ESB con wso2.
- b) Desarrollo de microservicios con Spring Boot, Docker y API Gateway.
- c) Diseño de componentes en AWS.
- d) Realizar otras actividades de investigación.

El área de cloud services es responsable de proponer migraciones a nube de los componentes de software de los sistemas actuales de los clientes.

El área de Cloud Services tiene las siguientes funciones:

- a) Diseñar e implementar nuevas arquitecturas de integración usando tecnología en nube.
- b) Soporte y mantenimiento de las nuevas arquitecturas implementadas.

- c) Proponer nuevos componentes que permitan mejorar el rendimiento de los sistemas actuales.
- d) Definir los recursos que soporten la implementación de los sistemas en cloud.

2.6 EXPERIENCIA PROFESIONAL REALIZADA EN LA ORGANIZACIÓN

Durante mi experiencia profesional como Arquitecto Cloud tuve las siguientes responsabilidades:

- A. Realizar el levantamiento de información de los documentos de transmisión para el control aduanero dentro de las unidades de negocios de TPP.
- B. Diseñar el modelo de datos que permitirá soportar la información de los documentos a transmitir.
- C. Diseñar los componentes de software de la nueva arquitectura de integración mediante el integrador empresarial wso2.
- D. Ejecutar las pruebas de estrés de los componentes de software de la arquitectura de integración.
- E. Diseñar los componentes físicos en nube para el despliegue de la nueva arquitectura de integración.

CAPITULO III - ACTIVIDADES DESARROLLADAS

3.1 SITUACION PROBLEMÁTICA

3.1.1 DEFINICION DEL PROBLEMA

Terminales Portuarios Peruanos SAC (TPP) en el año 2020 detecta que la arquitectura de integración que realiza las transmisiones de los documentos aduaneros de ingreso y salida al sistema electrónico de intercambio de documentos aduaneros (SEIDA) de SUNAT tiene **problemas de escalabilidad en tiempos picos de alta transacción.**

Principalmente se presenta fallas en los envíos de transmisión de documentos a SUNAT implicando un aumento en los costos por incumplimiento de las obligaciones tributarias y aduaneras por las constantes multas.

Se identifica debilidades de integración con nuevos canales origen desde donde se requiere enviar los documentos a transmitir al sistema SEIDA de SUNAT.

Poca capacidad del sistema legado de recuperación a errores de comunicación y conectividad con SUNAT dentro del límite de tiempo de transmisión de información.

No existe un mecanismo de recuperación de errores que permiten reintentar de forma automática la transmisión de documentos.

No existe un sistema de monitoreo en línea de los estados de los documentos transmitidos a SUNAT.

3.2 SOLUCION

3.2.1 OBJETIVO GENERAL

Implementar una arquitectura ESB mediante el integrador empresarial wso2 para la transmisión de documentos de terminales portuarios a SUNAT.

3.2.2 OBJETIVOS ESPECÍFICOS

- Diseñar un modelo de datos para el monitoreo de los estados e información de respuestas de los documentos aduaneros.
- Diseñar una arquitectura de integración usando el integrador empresarial wso2 para la transmisión de documentos a SUNAT.
- Desarrollar los procesos de transmisión usando flujos de negocios en ESB.
- Desarrollar una aplicación frontend para el monitoreo y validación de estados de los documentos transmitidos.

3.2.3 ALCANCE

El presente documento brinda información de las actividades realizadas como parte de la implementación de una arquitectura ESB mediante el integrador empresarial wso2 para la transmisión de documentos de terminales portuarios a SUNAT.

Alcance Organizacional

La implementación de la arquitectura de integración se relaciona con las siguientes unidades de negocios de TPP que realizan transmisión de documentos a SUNAT.

A.- Naves:

La unidad gestiona las operaciones de las naves para permitirles acceder a las instalaciones portuarias o retirarse de ellas.

B.- Recaladas:

La unidad gestiona las operaciones para el control de la salida e ingreso al territorio nacional de mercancías nacionales o nacionalizadas por cabotaje de carga con arribo a recintos marítimos internacionales.

C.- Contenedores:

La unidad se encarga de gestionar el monitoreo del estado de los contenedores de todas las actividades en tierra.

D.- Bill of Lading:

La unidad se encarga de generar los documentos de contrato para el transporte marítimo de importación o exportación.

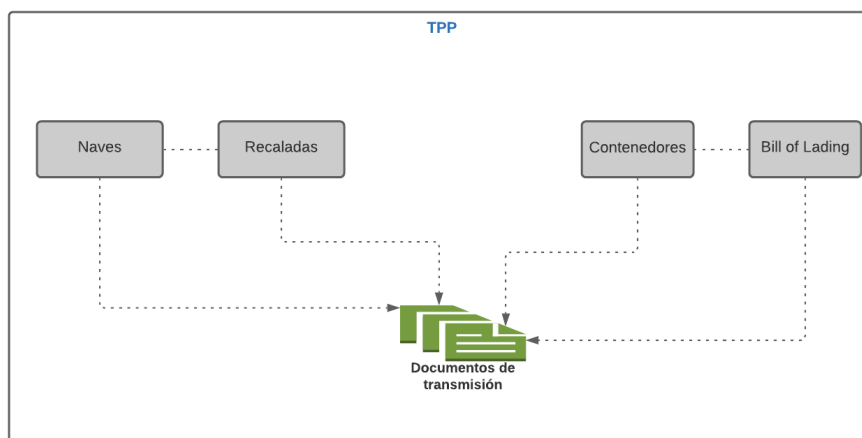


Figura 4-Alcance Organizacional en TPP (Fuente: Elaboración Propia)

3.2.4 ETAPAS Y METODOLOGIA

La gestión del proyecto se realizó usando metodología ágil Scrum con un tiempo estimado de 4 meses.

Se definieron los siguientes criterios para el proyecto.

a.- Total de Sprint: 4

b.- Duración por Sprint: 1 mes (4 semanas)

c.- Equipo Básico Scrum: Product Owner, Scrum Master y Equipo Scrum. Mi rol dentro del Equipo Scrum fue de Arquitecto de Software.

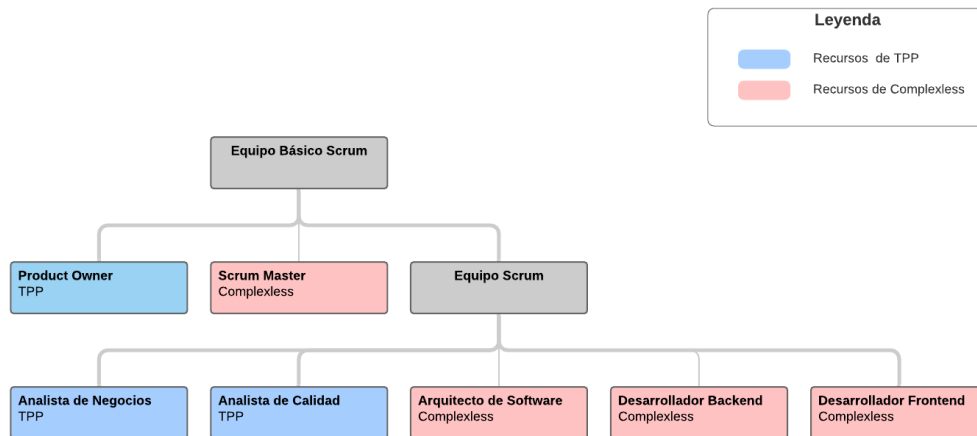


Figura 5-Equipo básico scrum del proyecto (Fuente: Elaboración Propia)

d.- Cronograma del proyecto.

El cronograma del proyecto contiene el tiempo programado por cada Sprint, las dependencias de tareas y los recursos involucrados.

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
PROYECTO - ARQUITECTURA DE INTEGRACION	85 días	lun 2/11/20	vie 26/02/21		
Sprint 1	20 días	lun 2/11/20	lun 30/11/20		
Sprint Planning	1 día	lun 2/11/20	lun 2/11/20		Product Owner;Scrum Master;Equipo Scrum
Identificación de Product Backlog Items	1 día	mar 3/11/20	mar 3/11/20	3	Equipo Scrum;Scrum Master
Ejecución de tareas del Sprint 1	18 días	mié 4/11/20	vie 27/11/20	4	Equipo Scrum
Reunión Sprint Review	0 días	vie 27/11/20	vie 27/11/20	5	Equipo Scrum;Product Owner;Scrum Master

Reunión Sprint Retrospective	0 días	lun 30/11/20	lun 30/11/20	5	Equipo Scrum;Product Owner;Scrum Master
Sprint 2	22 días	mar 1/12/20	mié 30/12/20	2	
Sprint Planning	1 día	mar 1/12/20	mar 1/12/20		Equipo Scrum;Product Owner;Scrum Master
Identificación de Product Backlog Items	1 día	mié 2/12/20	mié 2/12/20	9	Equipo Scrum;Scrum Master
Ejecución de tareas del Sprint 2	19 días	jue 3/12/20	mar 29/12/20	10	Equipo Scrum
Reunión Sprint Review	0 días	mar 29/12/20	mar 29/12/20	11	Equipo Scrum;Product Owner;Scrum Master
Reunión Sprint Retrospective	0 días	mar 29/12/20	mar 29/12/20	11	Equipo Scrum;Product Owner;Scrum Master
Sprint 3	20 días	lun 4/01/21	vie 29/01/21	8	
Sprint Planning	1 día	lun 4/01/21	lun 4/01/21		Equipo Scrum;Product Owner;Scrum Master
Identificación de Product Backlog Items	1 día	mar 5/01/21	mar 5/01/21	15	Equipo Scrum;Scrum Master
Ejecución de tareas del Sprint 3	17 días	mié 6/01/21	vie 29/01/21	16	Equipo Scrum
Reunión Sprint Review	0 días	vie 29/01/21	vie 29/01/21	17	Equipo Scrum;Product Owner;Scrum Master
Reunión Sprint Retrospective	0 días	vie 29/01/21	vie 29/01/21	17	Equipo Scrum;Product Owner;Scrum Master
Sprint 4	20 días	lun 1/02/21	vie 26/02/21	14	
Sprint Planning	1 día	lun 1/02/21	lun 1/02/21		Equipo Scrum;Product Owner;Scrum Master
Identificación de Product Backlog Items	1 día	mar 2/02/21	mar 2/02/21	21	Equipo Scrum;Scrum Master
Ejecución de tareas del Sprint 4	17 días	mié 3/02/21	jue 25/02/21	22	Equipo Scrum
Reunión Sprint Review	0 días	vie 26/02/21	vie 26/02/21	23	Equipo Scrum;Product Owner;Scrum Master
Reunión Sprint Retrospective	0 días	vie 26/02/21	vie 26/02/21	23	Equipo Scrum;Product Owner;Scrum Master

e.- Product Backlog.

Resultado del trabajo del Product Owner con los diferentes interesados (usuarios de las unidades de negocio de TPP) antes del inicio de los sprints se presentó la lista de características priorizadas que contiene descripciones sobre todo lo que se desea del producto q se va a implementar.

Tabla 2. Product Backlog del proyecto.

ID	Como	Necesito	Para	Criterios de Aceptación
1	Usuario	Enviar operaciones de numeración y rectificación de "Ingreso y recepción de mercancías - IRM"	Cumplir control aduanero de ingresos de mercancías de SUNAT.	<p>1.1 Que la transmisión tenga una respuesta exitosa de la Aduana.</p> <p>1.2 Que la transmisión se realice por nueva arquitectura ESB usando WSO2.</p> <p>1.3 Que el plazo de transmisión sea de 24 horas desde el ingreso del último bulto.</p>
2	Usuario	Enviar operaciones de "Registro de salida de mercancías - RSA"	Cumplir control aduanero de salidas de mercancías de SUNAT.	<p>2.1 Que la transmisión tenga una respuesta exitosa de la Aduana.</p> <p>2.2 Que la transmisión se realice por nueva arquitectura ESB usando WSO2.</p> <p>2.3 Que el plazo de transmisión sea 30 minutos desde el registro de la segunda pesada del camión.</p>
3	Usuario	Monitorear operaciones de IRM y RSA	Cumplir control aduanero de ingresos y salidas de mercancías de SUNAT.	<p>3.1 Que exista una aplicación web con un módulo para "Transmisiones" y opción para "Ingreso y recepción de mercancías (IRM)". <u>Ver prototipo de pantalla de monitoreo de IRM.</u></p> <p>3.2 Que exista una aplicación web con un módulo para "Transmisiones" y opción para "Registro de salida de mercancías (RSA)". <u>Ver prototipo de pantalla de monitoreo de RSA.</u></p>

Se presentaron los prototipos para realizar el monitoreo de las transmisiones.

Pantalla de Monitoreo de IRM: Se muestra información sobre cada transmisión de IRM dividido en tres secciones: Información sobre el evento (BL, Recalada, Fecha y hora de ingreso), información sobre la transacción (Tipo transmisión, Tiempo transcurrido, Tiempo restante, Estado Transmisión y Mensaje de Error) y el detalle de la transmisión. También se agrega los filtros de búsqueda ubicados en la cabecera.

Pantalla de Monitoreo de RSA: Se muestra información sobre cada transmisión de RSA dividido en tres secciones: Información sobre el evento (Contenedor, Recalada, Fecha y hora de salida), información sobre la transacción (Tipo transmisión, Tiempo transcurrido, Tiempo restante, Estado Transmisión y Mensaje de Error) y el detalle de la transmisión.

Filtros de cabecera		contenedores		▼					
		CARGA SUELTA	<input type="text"/>						
		CARGA RODANTE	<input type="text"/>	▼					
IRM									
INFORMACIÓN SOBRE EL EVENTO			INFORMACIÓN SOBRE LA TRANSACCIÓN						
BL	Recalada	Fecha y hora último ingreso	Tipo Transmisión	Tiempo transcurrido	Tiempo Restante	Estado de transmisión	Mensaje de error	Acción	
XXXXXXX	HAMMONIA FORTUNA 0565	02/09 03:00am		5 hrs y 30 min	18 hrs y 30 min	Sin respuesta	-		
YYYYYYY	HAMMONIA FORTUNA 0565	02/09 02:59am		5 hrs y 31 min	18 hrs y 29 min	Con error	"el número de detalle no existe en el manifiesto de carga"		
Información de		Bill of Lading: HMDU65498756		Cant. Bultos: 3,500		Condición carga: FCL			
		Manifiesto: 118-2020-4568		Detalle: 56		Peso Bruto: 26,500			
ZZZZZZZ	COSCO YANTIAN 022E	02/09 03:10am		5 hrs y 20 min	18 hrs y 40 min	pendiente de envío	-		
HHHHHHH	COSCO YANTIAN 022E	02/09 03:11am		5 hrs y 19 min	18 hrs y 41 min	pendiente de envío	-		

Figura 6-Prototipo de pantalla de monitoreo IRM (Fuente: Elaboración Propia)

RSA									
INFORMACIÓN SOBRE EL EVENTO			INFORMACIÓN SOBRE LA TRANSACCIÓN						
Contenedor	Recalada	Fecha y hora de Salida	Tipo Transmisión	Tiempo transcurrido	Tiempo Restante	Estado de transmisión	Mensaje de error		
GLDU4978568	HAMMONIA FORTUNA 0565	02/09 03:00am		25 min	5 min	Sin respuesta	-		
HMCU9876543	HAMMONIA FORTUNA 0565	02/09 02:59am		26 min	4 min	Con error	"La cantidad de bultos no coincide con el manifiesto de carga"		
Información de campos transmitidos		Bill of Lading: HMDU65498756		Ticket de puerto: 456456		RUC transportista: 205076548974			
		Manifiesto: 118-2020-4568		Placa: AUY689		Tara camión: 4,500			
		Detalle: 56		Brevete: Q43578962		Peso Bruto: 26,500			
		Tipo de carga: FCL		Nro. Volante: 456987		Peso Neto: 23,500			
		Cant. Bultos: 3500		Nro. DAM: 118-2020-65478		Régimen: 10			
		Aut. Retiro: 654987		Estado levante: Levante aut.					
AMCU9876541	COSCO YANTIAN 022E	02/09 03:10am		15 min	15 min	pendiente de envío	-		
AMCU9876541	COSCO YANTIAN 022E	02/09 03:11am		14 min	16 min	pendiente de envío	-		

Figura 7-Prototipo de pantalla de monitoreo RSA (Fuente: Elaboración Propia)

f.- Sprint 1.

Como primera actividad se programó la reunión de Sprint Planning con los miembros del equipo básico scrum. El Product Owner explicó detalladamente los ítems del Product Backlog, se absolvieron las consultas y dudas del Equipo Scrum.

El Equipo Scrum definió las siguientes épicas e historias de usuarios para el proyecto. Se resalta en naranja las Historia de Usuarios que como Arquitecto de Software se tuvo participación.

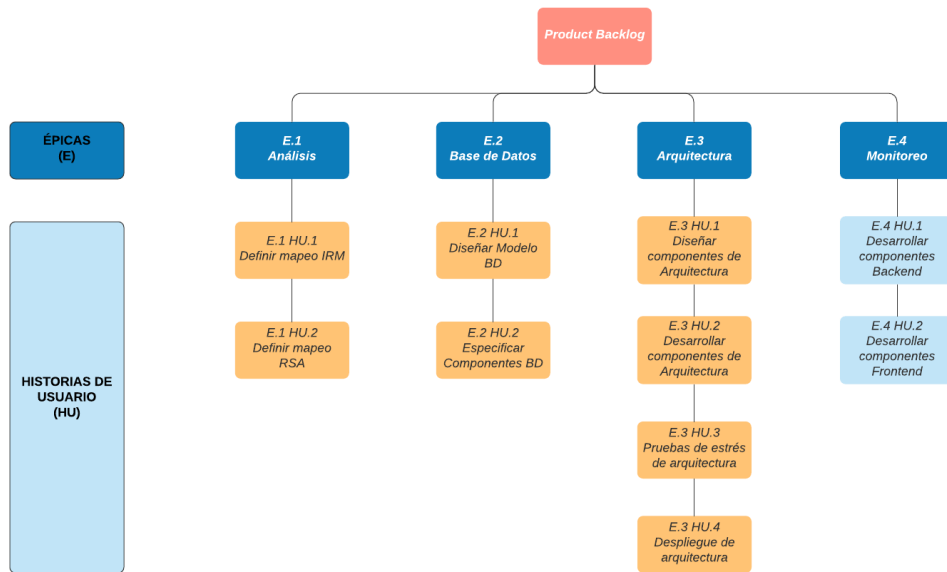


Figura 8-Épicas e Historias de Usuario del Proyecto (Fuente: Elaboración Propia)

El Scrum master y el Equipo Scrum definieron las tareas del Sprint backlog para el Sprint 1.

HISTORIAS DE USUARIO (HU)	TO DO	RECURSOS
E.1 HU.1 Definir mapeo IRM	E.1 HU.1 T.1 Identificar campos de negocio de IRM	Arquitecto de Software Analista de negocios
	E.1 HU.1 T.2 Mapear campos de negocio con plantilla IRM	Arquitecto de Software Analista de negocios
E.1 HU.2 Definir mapeo RSA	E.1 HU.2 T.1 Identificar campos de negocio de RSA	Arquitecto de Software Analista de negocios
	E.1 HU.2 T.2 Mapear campos de negocio con plantilla RSA	Arquitecto de Software Analista de negocios
E.2 HU.1 Diseñar Modelo BD	E.2 HU.1 T.1 Diseñar modelo de datos para Transmisiones	Arquitecto de Software
	E.2 HU.1 T.2 Generar scripts de base de datos para Transmisiones	Arquitecto de Software
E.2 HU.2 Especificar Componentes BD	E.2 HU.2 T.1 Generar diccionario de datos	Arquitecto de Software

Figura 9-Sprint Backlog del Sprint 01 (Fuente: Elaboración Propia)

Ejecución de tareas.

Identificar campos de negocio de IRM

En la primera sesión de trabajo entre el Analista de negocios y el Arquitecto de Software, el Analista de Negocios confirmó que la plantilla para Ingreso y Recepción de mercancías se encuentran en la siguiente ruta del Portal de SUNAT:

Operatividad Aduanera – Manifiesto de Ingreso

<https://www.sunat.gob.pe/operatividadaduanera/fast/estructuras/formato-Mani-Ingreso.html>

La plantilla se encuentra en un formato de intercambio de datos electrónicos denominado CUSCAR-XML.

Tabla 3. Plantilla de Ingreso y recepción de mercancías

```
<?xml version="1.0" encoding="UTF-8"?>
<DeclarationMetaData xmlns="urn:wco:datamodel:PE:DocumentMetaData:1"
xmlns:ds="urn:wco:datamodel:PE:MetaData_DS:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:wco:datamodel:PE:DocumentMetaData:1 DeclarationMetaData_PE_1p0.xsd">
  <WCODataModelVersionCode>3.7</WCODataModelVersionCode>
  <WCOTypeName>DEC</WCOTypeName>
  <ResponsibleCountryCode>PE</ResponsibleCountryCode>
  <ResponsibleAgencyName>SUNAT</ResponsibleAgencyName>
  <FunctionalDefinition/>
  <Declaration xmlns="urn:wco:datamodel:PE:Declaration:1"
xmlns:ds="urn:wco:datamodel:PE:Declaration_DS:1" xmlns:eds="urn:wco:datamodel:PE:Declaration_EDS:1">
    <FunctionCode/>
    <FunctionalReferenceID/>
    <IssueDateTime>
      <ds:DateTimeString formatCode="204"/>
    </IssueDateTime>
    <TypeCode>785</TypeCode>
    <VersionID>1.0</VersionID>
    <Submitter>
      <ID schemeID="4">20507646051</ID>
      <RoleCode>31</RoleCode>
    </Submitter>
    <DeclarationOffice>
      <ID>118</ID>
    </DeclarationOffice>
    <AdditionalDocument>
      <CategoryCode>01</CategoryCode>
      <ID/>
      <IssueDateTime>
        <ds:DateTimeString formatCode="602"/>
      </IssueDateTime>
      <issueLocationID>118</issueLocationID>
      <TypeCode>785</TypeCode>
    </AdditionalDocument>
    <BorderTransportMeans>
      <ModeCode>1</ModeCode>
    </BorderTransportMeans>
    <Consignment>
      <SequenceNumeric/>
```

```

<TotalPackageQuantity/>
<TotalGrossMassMeasure/>
<AdditionalInformation>
  <StatementCode>61</StatementCode>
  <StatementTypeCode>ACH</StatementTypeCode>
</AdditionalInformation>
<Control>
  <LimitDateTime>
    <ds:DateTimeString formatCode="204"/>
  </LimitDateTime>
</Control>
<TransportContractDocument>
  <ID/>
</TransportContractDocument>
</Consignment>
</Declaration>
</DeclarationMetaData>

```

En la segunda y tercera sesión de trabajo se realizaron el análisis de los campos de negocio que se identificaron en la plantilla de IRM.

Tabla 4. Campos de negocio para el Ingreso y recepción de mercancías

Campo	Tipo	Formato	Descripción
Código de transacción	String		Código de transacción de IRM. Numeración: 0121 Rectificación: 0122
Número de referencia del emisor	String		RUC del emisor del documento Ejemplo: 2018850010
Fecha y hora de generación	Date	yyyyMMddHHm mss	Fecha y hora que se generó el documento de transporte.
Identificador del emisor	String		Código de identificación del emisor. Ejemplo: 2018850010
Número del Manifiesto	Integer		Número del manifiesto del documento de transporte.
Año del Manifiesto	Integer	yyyy	Año del manifiesto del documento de transporte.
Secuencia del Documento de transporte	Integer		Número de Secuencia del Documento de transporte
Bultos Totales Recibidos	Integer		Bultos Totales Recibidos de ingreso

Campo	Tipo	Formato	Descripción
Peso Total Recibido	Integer		Peso Total Recibido de ingreso
Fecha de la operación asociada	Date	yyyyMMddHHm mss	Fecha de la operación asociada al ingreso
Numero de documento de transporte	String		Numero de documento o guía de transporte.

Mapear campos de negocio con plantilla IRM

En la cuarta sesión se hizo el análisis de identificar que campos de negocio corresponden a las etiquetas de la plantilla. Se realizó el siguiente mapeo:

Tabla 5. Mapeo de campos de negocio con las etiquetas de la plantilla de IRM

Campo	Etiqueta en xml
Código de transacción	<FunctionCode></FunctionCode>
Número de referencia del emisor	<FunctionalReferenceID></FunctionalReferenceID>
Fecha y hora de generación	<IssueDateTime><ds:DateTimeStringformatCode="204"></ds:DateTimeStringformatCode></IssueDateTime>
Identificador del emisor	<Submitter><ID schemeID="4"></ID></Submitter>
Número del Manifiesto	<AdditionalDocument><ID></ID></AdditionalDocument>
Año del Manifiesto	<AdditionalDocument><IssueDateTime><ds:DateTimeStringformatCode="602"></ds:DateTimeString></IssueDateTime></AdditionalDocument>
Secuencia del Documento de transporte	<Consignment><SequenceNumeric></SequenceNumeric></Consignment>
Bultos Totales Recibidos	<Consignment><TotalPackageQuantity></TotalPackageQuantity></Consignment>
Peso Total Recibido	<Consignment><TotalGrossMassMeasure></TotalGrossMassMeasure></Consignment>

Campo	Etiqueta en xml
Fecha de la operación asociada	<Consignment><Control><LimitDateTime><ds:DateTimeString formatCode="204"></ds:DateTimeString></LimitDateTime></Control></Consignment>
Número de documento de transporte	<Consignment><TransportContractDocument><ID></ID></TransportContractDocument></Consignment>

Identificar campos de negocio de RSA

En la quinta sesión de trabajo entre el Analista de negocios y el Arquitecto de Software, el Analista de Negocios confirmó que la plantilla para Registro de salida de mercancías se encuentra en la siguiente ruta del Portal de SUNAT:

Operatividad Aduanera – Manifiesto de Salida

<https://www.sunat.gob.pe/operatividadaduanera/fast/estructuras/formato-Mani-Salida.html>

La plantilla se encuentra en un formato de intercambio de datos electrónicos denominado CUSCAR-XML.

Tabla 6. Plantilla de Registro de salida de mercancías

```
<?xml version="1.0" encoding="UTF-8"?>
<urn:SalidaAlmacenRequest
xmlns:urn="urn:wco:datamodel:pe:gob:sunat:controladuanero:ingreso:reconfisico:DeclaracionReconocimientoFisicoRequest:1" xmlns:urn1="urn:wco:datamodel:WCO:Declaration_DS:1">
  <urn:Submitter>
    <urn:RoleCode>31</urn:RoleCode>
  </urn:Submitter>
  <urn:DeclarationOfficeID/>
  <urn:IssueDateTime>
    <urn1:DateTimeString formatCode="602"/>
  </urn:IssueDateTime>
  <urn:GovernmentProcedure>
    <urn:CurrentCode/>
  </urn:GovernmentProcedure>
  <urn:ID/>
  <urn:GoodsShipment>
    <urn:Consignment>
      <urn:AdditionalInformation>
        <urn:Content/>
        <urn:StatementTypeCode>AAX</urn:StatementTypeCode>
      </urn:AdditionalInformation>
      <urn:BorderTransportMeans>
        <urn:ID/>
      </urn:BorderTransportMeans>
      <urn:TransportEquipment>
        <urn:ID/>
        <urn:Seal>
          <urn:ID/>
        </urn:Seal>
      </urn:Seal>
    </urn:Consignment>
  </urn:GoodsShipment>
</urn:SalidaAlmacenRequest>
```

```

</urn:TransportEquipment>
</urn:Consignment>
<urn:GoodsMeasure>
  <urn:GrossMassMeasure/>
</urn:GoodsMeasure>
<urn:Warehouse>
  <urn:DepartureDateTime>
    <urn1:DateTimeString formatCode="204"/>
  </urn:DepartureDateTime>
</urn:Warehouse>
<urn:AdditionalInformation>
  <urn:Content/>
  <urn:StatementCode>KIL</urn:StatementCode>
  <urn:StatementTypeCode>AAQ</urn:StatementTypeCode>
</urn:AdditionalInformation>
<urn:AdditionalInformation>
  <urn:Content/>
  <urn:StatementTypeCode>AFB</urn:StatementTypeCode>
</urn:AdditionalInformation>
<urn:AdditionalInformation>
  <urn:StatementCode/>
  <urn:StatementTypeCode>AID</urn:StatementTypeCode>
</urn:AdditionalInformation>
<urn:AdditionalInformation>
  <urn:StatementCode/>
  <urn:StatementTypeCode>LOI</urn:StatementTypeCode>
</urn:AdditionalInformation>
</urn:GoodsShipment>
</urn:SalidaAlmacenRequest>

```

En la sexta y séptima sesión de trabajo se realizaron el análisis de los campos de negocio que se identificaron en la plantilla de RSA.

Tabla 7. Campos de negocio para el Registro de salida de mercancías

Campo	Tipo	Formato	Descripción
Código aduana de la declaración	String		Código aduana de la declaración de salida
Año de la declaración	Integer		Año de la declaración de salida
Código de régimen de la declaración	String		Código de régimen de la declaración de salida
Número de la declaración	String		Número de la declaración de salida
Licencia del conductor	String		Número de licencia o breveté del conductor
Número de placa	String		Número de placa del vehículo de salida

Campo	Tipo	Formato	Descripción
Número de equipamiento	String		Número del manifiesto del documento de transporte.
Número de precinto	String		Número de precinto del contenedor
Peso bruto	Integer		Peso bruto de la mercancía
Fecha de salida del almacén	Date	yyyyMMddHHmmss	Fecha de salida del almacén de la mercancía
Cantidad total de bultos	Integer		Cantidad total de bultos de la mercancía

Mapear campos de negocio con plantilla RSA

En la octava sesión se hizo el análisis de identificar que campos de negocio corresponden a las etiquetas de la plantilla. Se realizó el siguiente mapeo:

Tabla 8. Mapeo de campos de negocio con las etiquetas de la plantilla de RSA

Campo	Etiqueta en xml
Código aduana de la declaración	<urn:DeclarationOfficeID></urn:DeclarationOfficeID>
Año de la declaración	<urn:IssueDateTime><urn1:DateTimeString formatCode="602"></urn1:DateTimeString></urn:IssueDateTime>
Código de régimen de la declaración	<urn:GovernmentProcedure><urn: CurrentCode></urn: CurrentCode></urn:GovernmentProcedure>
Número de la declaración	<urn:ID></urn:ID>
Licencia del conductor	<urn:GoodsShipment><urn:Consignment><urn: AdditionalInformation><urn:Content></urn:Content></urn: AdditionalInformation></urn:Consignment></urn:GoodsShipment>
Número de placa	<urn:GoodsShipment><urn:Consignment><urn: BorderTransportMeans><urn:ID></urn:ID></urn: BorderTransportMeans></urn:Consignment></urn:GoodsShipment>
Número de equipamiento	<urn:GoodsShipment><urn:Consignment><urn: TransportEquipment><urn:ID></urn:ID></urn: TransportEquipment></urn:Consignment></urn:GoodsShipment>
Número de precinto	<urn:GoodsShipment><urn:Consignment><urn: TransportEquipment><urn: Seal><urn:ID></urn:ID></urn: Seal></urn: TransportEquipment></urn:Consignment></urn:GoodsShipment>

Campo	Etiqueta en xml
Peso bruto	<urn:GoodsShipment><urn:GoodsMeasure><urn:GrossMassMeasure</urn:GrossMassMeasure></urn:GoodsMeasure></urn:GoodsShipment>
Fecha de salida del almacén	<urn:GoodsShipment><urn:Warehouse><urn:DepartureDateTime><urn1:DateTimeString formatCode="204"></urn1:DateTimeString></urn:DepartureDateTime></urn:Warehouse></urn:GoodsShipment>
Cantidad total de bultos	<urn:GoodsShipment><urn:AdditionalInformation><urn:Content></urn:Content></urn:AdditionalInformation></urn:GoodsShipment>

Diseñar modelo de datos para Transmisiones

El arquitecto de software diseñó a través de un modelador de Entidad Relación el modelo de datos que dará soporte a las configuraciones, transmisiones y reintentos de las operaciones de IRM y RSA.

Los nombres de los objetos de la base de datos son nombrados usando los estándares de Base de Datos del **Anexo 02**.

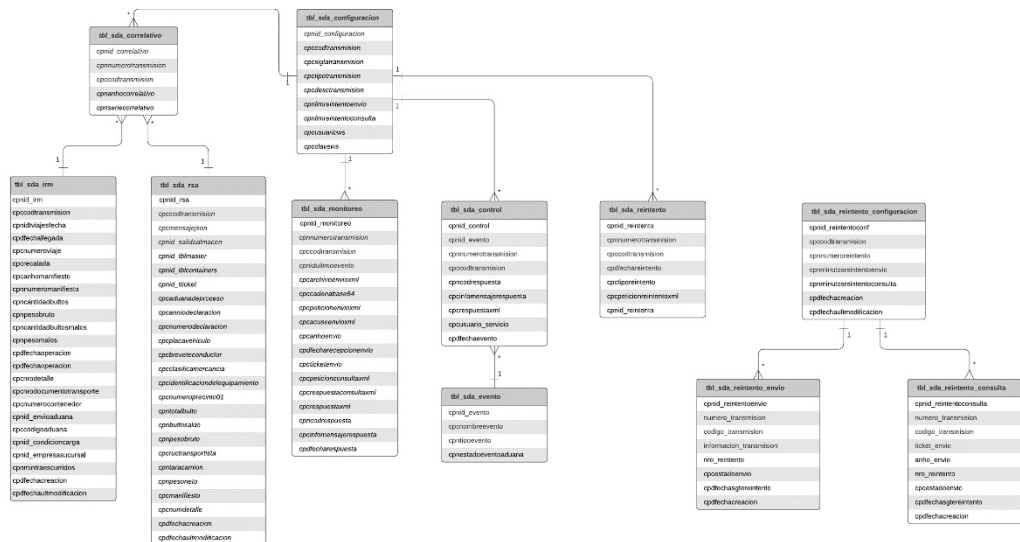


Figura 10-Modelo de datos para Transmisiones (Fuente: Elaboración Propia)

Generar scripts de base de datos para Transmisiones

Los objetos que se crearán para el soporte de las configuraciones, transmisiones y reintentos son: Tablas, Funciones, Triggers.

Tabla 9. Scripts de generación de objetos de la Base de Datos de Transmisiones

```

/*
-----
PROYECTO TRANSMISIONES
-----
*/

/*
-----
Creacion de base de datos y permisos
-----
*/
CREATE DATABASE transmisiones WITH OWNER = postgres ENCODING = 'UTF8'
TABLESPACE=transmisiones_sde_dict;

ALTER DATABASE transmisiones SET search_path="$user", public, sde;
GRANT ALL ON DATABASE transmisiones TO public;
GRANT ALL ON DATABASE transmisiones TO postgres;

/*
-----
Creacion de tablas
-----
*/

CREATE TABLE public.tbl_sda_configuracion
(
    cpnid_configuracion integer NOT NULL DEFAULT
nextval('tbl_sda_configuracion_cpnid_configuracion_seq'::regclass),
    cpccodtransmision character(4) NOT NULL,
    cpsiglatransmision character varying(100) NOT NULL,
    cpctipotransmision character varying(100) NOT NULL,
    cpccdescrtransmision character varying(100) NOT NULL,
    cpctablatransmision character varying(50) NOT NULL,
    cpctabladetalletransmision character varying(500) NOT NULL,
    cpnlimiteminutos integer NOT NULL,
    cpnumbralalerta integer NOT NULL,
    cpnumbralcritico integer NOT NULL,
    cpccorreoalertausuario character varying(500) NOT NULL,
    cpnlimreintentoenvio integer NOT NULL,
    cpnlimreintentoconsulta integer NOT NULL,
    cpcusuariows character varying(100) COLLATE NOT NULL,
    cpclavews character varying(50) COLLATE NOT NULL,
    cpnestadoconfig smallint NOT NULL DEFAULT 1,
    cpcusuario_servicio character varying(50) NOT NULL,
    cpdfechacreacion timestamp without time zone NOT NULL DEFAULT CURRENT_TIMESTAMP,
    cpdfechaultmodificacion timestamp without time zone NOT NULL DEFAULT CURRENT_TIMESTAMP,
    cpccodtransaccionoma character varying(4) NOT NULL,
    cpnlimiteplazoenvio integer NOT NULL DEFAULT 0,
    CONSTRAINT pk_tbl_sda_configuracion PRIMARY KEY (cpnid_configuracion),
    CONSTRAINT tbl_sda_configuracion_cpccodtransmision_key UNIQUE (cpccodtransmision)
);

CREATE TABLE public.tbl_sda_correlativo
(
    cpnid_correlativo bigint NOT NULL DEFAULT nextval('tbl_sda_correlativo_cpnid_correlativo_seq'::regclass),
    cpnnumerotransmision bigint NOT NULL,
    cpccodtransmision character(4) NOT NULL,
    cpnanhocorrelativo smallint NOT NULL,
    cpnseriecorrelativo integer NOT NULL,
    cpdfechacreacion timestamp without time zone NOT NULL DEFAULT CURRENT_TIMESTAMP,

```

```

CONSTRAINT transmision_correlativo_pkey PRIMARY KEY (cpnid_correlativo)
);

CREATE TABLE public.tbl_sda_evento
(
cpnid_evento integer NOT NULL,
cpcnombreevento character varying(50) NOT NULL,
cpntipoevento smallint NOT NULL,
cpnestadoeventoaduana smallint NOT NULL,
CONSTRAINT pk_tbl_sda_evento PRIMARY KEY (cpnid_evento)
);

CREATE TABLE public.tbl_sda_control
(
cpnnumero transmision bigint NOT NULL,
cpccod transmision character(4) NOT NULL,
cpnid_evento integer NOT NULL,
cpncod respuesta integer,
cpcifomensaje respuesta text,
cpcrespuesta xml text,
cpusuario_servicio character varying(50),
cpdfechaevento timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
cpnid_control bigint NOT NULL DEFAULT nextval('tbl_sda_control_cpnid_control_seq'::regclass),
CONSTRAINT pk_tbl_sda_control PRIMARY KEY (cpnid_control),
CONSTRAINT fk_cpccod transmision FOREIGN KEY (cpccod transmision)
REFERENCES public.tbl_sda_configuracion (cpccod transmision) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION,
CONSTRAINT fk_cpnid_evento FOREIGN KEY (cpnid_evento)
REFERENCES public.tbl_sda_evento (cpnid_evento) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
);

CREATE TABLE public.tbl_sda_monitoreo
(
cpnnumero transmision bigint NOT NULL,
cpccod transmision character(4) NOT NULL,
cpnidultimoevento integer,
cpdfechaultimoevento timestamp without time zone,
cparchivoenvio xml text,
cpccadenabase64 text,
cpcpeticionenvio xml text,
cpcacuseenvio xml text,
cpcanhoenvio character(4),
cpdfecharecepcionenvio timestamp without time zone,
cpcticketenvio character varying(20),
cpcpeticionconsulta xml text,
cpcrespuestaconsulta xml text,
cpcrespuesta xml text,
cpncod respuesta integer,
cpcifomensaje respuesta text,
cpdfecha respuesta timestamp without time zone,
cpndeshabilitar intento smallint DEFAULT 0,
cpusuario_servicio character varying(50) COLLATE,
cpdfechacreacion timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
cpdfechaultmodificacion timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
cpnestadoregistro smallint DEFAULT 1,
cpnid_envioaduana bigint,
cpnid_monitoreo bigint NOT NULL DEFAULT nextval('tbl_sda_monitoreo_cpnid_monitoreo_seq'::regclass),
CONSTRAINT pk_tbl_sda_monitoreo PRIMARY KEY (cpnid_monitoreo),
CONSTRAINT tbl_sda_monitoreo_cpnnumero transmision_cpccod transmision_key UNIQUE
(cpnnumero transmision, cpccod transmision),

```

```

CONSTRAINT fk_cpccodtransmision FOREIGN KEY (cpccodtransmision)
  REFERENCES public.tbl_sda_configuracion (cpccodtransmision) MATCH SIMPLE
  ON UPDATE NO ACTION
  ON DELETE NO ACTION
);

CREATE TABLE public.tbl_sda_irm
(
  cpnid_irm bigint NOT NULL DEFAULT nextval('tbl_sda_irm_cpnid_irm_seq'::regclass),
  cpccodtransmision character(4) NOT NULL,
  cpcmensajejson json NOT NULL,
  cpnidviasjesfecha integer,
  cpdfechallegada timestamp without time zone,
  cpcnumeroviaje character varying(200),
  cpcrecalada character varying(50),
  cpcanhomanifiesto character(4),
  cpnnumeromanifiesto integer,
  cpncantidadbultos integer,
  cpnpesobruto numeric(13,3),
  cpncantidadbultosmalos integer,
  cpnpesomalos numeric(13,3),
  cpdfechaoperacion timestamp without time zone,
  cpnid_conocimiento bigint,
  cpcnrodetalle character varying(5),
  cpcnrodocumentotransportemaster character varying(34),
  cpcnrodocumentotransporte character varying(34),
  cnpasarlcl integer,
  cpctipo character varying(60),
  cpccierrebl character varying(60),
  cpnid_contenedor bigint,
  cpcnumerocontenedor character varying(60),
  cpnindicasini integer,
  cpnindicasinicontacto integer,
  cpcnroactasini character varying(100),
  cpnid_envioaduana bigint,
  cpceestadoenvio character varying(60),
  cpcinformacionmensaje character varying(200),
  cpncantidadbultosmanifestado integer,
  cpccodigoaduana character(4),
  cpnid_tipoenvioconfigura bigint,
  cpnid_condicioncarga bigint,
  cpnid_empresasucursal bigint,
  cpnmintranscurridos integer,
  cpcemisor character varying(50),
  cpctransmisor character varying(50),
  cpcusuarioservicio character varying(50),
  cpdfechacreacion timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
  cpdfechaultmodificacion timestamp without time zone,
  CONSTRAINT tbl_sda_irm_pkey PRIMARY KEY (cpnid_irm),
  CONSTRAINT fk_irm_codtransmision FOREIGN KEY (cpccodtransmision)
    REFERENCES public.tbl_sda_configuracion (cpccodtransmision) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
);

CREATE TABLE public.tbl_sda_rsa
(
  cpnid_rsa bigint NOT NULL DEFAULT nextval('tbl_sda_rsa_cpnid_rsa_seq'::regclass),
  cpccodtransmision character(4) NOT NULL,
  cpcmensajejson json NOT NULL,
  cpnid_salidaalmacen bigint,
  cpnid_blmastersalidaalmacen bigint,
  cpnid_tblmaster bigint,

```

```

cpnid_tblcontainers bigint,
cpnid_tautorizacion bigint,
cpnid_tticket bigint,
cpcaduanadeproceto character varying(200),
cpcannioddeclaracion character varying(10),
cpccodigoregimendeclaracion character varying(100),
cpcnumerodeclaracion character varying(100),
cpcplacavehiculo character varying(50),
cpcbreveteconductor character varying(50),
cpcclasificamercancia character varying(10),
cpcidentificaciondeequipamiento character varying(50),
cpcnumeroprecinto01 character varying(50),
cpntotalbulto integer,
cpnbultosaldo numeric(13,3),
cpnpesobruto numeric(13,3),
cpdfechasalida timestamp without time zone,
cpcdescripcionmercancia character varying(200),
cpnvalidapesobulto integer,
cpnsalidaparcial integer,
cpncodigoestadoenvio integer,
cpcnumerocorrelativodeclaracion character varying(100),
cpcnumautorizacion character varying(100),
cpcnumdocuadua character varying(100),
cpcobservacion character varying(200),
cpcdescripcionrespuesta character varying(200),
cpnid_envioaduana bigint,
cpnid_empresasucursal bigint,
cpnid_tipoenvioconfigura bigint,
cpnid_condicioncarga bigint,
cpnmintranscurridos integer,
cpcusuarioservicio character varying(50),
cpdfechacreacion timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
cpdfechaultmodificacion timestamp without time zone,
cpcrecalada character varying,
cpcnumbld character varying,
cpnnumerovolante integer,
cpcnumerodam character varying,
cpcrucrtransportista character varying,
cpntaracamion numeric(13,3),
cpnpesoneto numeric(13,3),
cpcmanifiesto character varying,
cpcnumdetalle character varying,
CONSTRAINT tbl_sda_rsa_pkey PRIMARY KEY (cpnid_rsa),
CONSTRAINT fk_rsa_codtransmision FOREIGN KEY (cpcodtransmision)
REFERENCES public.tbl_sda_configuracion (cpcodtransmision) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
);

CREATE TABLE public.tbl_sda_reintento
(
cpnnumerotransmision bigint NOT NULL,
cpcodtransmision character(4) NOT NULL,
cpdfechareintento timestamp without time zone,
cpctiporeintento character varying(50),
cpcpeticionreintentoxml text,
cpnid_reintento bigint NOT NULL DEFAULT nextval('tbl_sda_reintento_cpnid_reintento_seq)::regclass),
CONSTRAINT pk_tbl_sda_reintento PRIMARY KEY (cpnid_reintento),
CONSTRAINT fk_cpccodtransmision FOREIGN KEY (cpcodtransmision)
REFERENCES public.tbl_sda_configuracion (cpcodtransmision) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
);

```

```

CREATE TABLE public.tbl_sda_reintento_configuracion
(
  cpnid_reintentoconf bigint NOT NULL DEFAULT
nextval('tbl_sda_reintento_configuracion_cpnid_reintentoconf_seq'::regclass),
  cpccodtransmision character(4),
  cpnnumero_reintento integer NOT NULL,
  cpnminutos_reintentoenvio integer NOT NULL,
  cpnminutos_reintentoconsulta integer NOT NULL,
  cpdfechacreacion timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
  cpdfechaultmodificacion timestamp without time zone,
  CONSTRAINT tbl_sda_reintento_configuracion_pkey PRIMARY KEY (cpnid_reintentoconf),
  CONSTRAINT tbl_sda_reintento_conf_cpccodtransmision_key UNIQUE (cpccodtransmision,
cpnnumero_reintento)
);

```

```

CREATE TABLE public.tbl_sda_reintento_envio
(
  cpnid_reintentoenvio bigint NOT NULL DEFAULT
nextval('tbl_sda_reintento_envio_cpnid_reintentoenvio_seq'::regclass),
  numero_transmision character varying(20),
  codigo_transmision character varying(4),
  informacion_transmision character varying(4000),
  nro_reintento character varying(5),
  cpcestadoenvio character varying(10) NOT NULL,
  cpdfechasgtreintento timestamp without time zone NOT NULL,
  cpdfechacreacion timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
  CONSTRAINT tbl_sda_reintento_envio_pkey PRIMARY KEY (cpnid_reintentoenvio)
);

```

```

CREATE TABLE public.tbl_sda_reintento_consulta
(
  cpnid_reintentoconsulta bigint NOT NULL DEFAULT
nextval('tbl_sda_reintento_consulta_cpnid_reintentoconsulta_seq'::regclass),
  numero_transmision character varying(20) NOT NULL,
  codigo_transmision character varying(4) NOT NULL,
  ticket_envio character varying(10),
  anho_envio character(4),
  nro_reintento character varying(5) NOT NULL,
  cpcestadoenvio character varying(10) NOT NULL,
  cpdfechasgtreintento timestamp without time zone NOT NULL,
  cpdfechacreacion timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
  CONSTRAINT tbl_sda_reintento_consulta_pkey PRIMARY KEY (cpnid_reintentoconsulta)
);

```

/*

Creacion de funciones y triggers para IRM

```

*/
CREATE OR REPLACE FUNCTION funsda_irm_insert()
  RETURNS trigger AS
$$
BEGIN
  INSERT INTO "tbl_sda_monitoreo" ( "cpnnumerotransmision", "cpccodtransmision", "cpcusuario_servicio",
"cpdfechaultmodificacion", "cpnestadoregistro","cpnid_envioaduana")
  VALUES(NEW."cpnid_irm",NEW."cpccodtransmision", 'SERV_FACADE', CURRENT_TIMESTAMP, 1,
NEW."cpnid_envioaduana");
  INSERT INTO "tbl_sda_correlativo" ( "cpnnumerotransmision", "cpccodtransmision", "cpnanhocorrelativo")
  VALUES(NEW."cpnid_irm",NEW."cpccodtransmision", DATE_PART('year', CURRENT_DATE)::INTEGER);
RETURN NEW;
END;

```

```

$$
LANGUAGE 'plpgsql';

CREATE TRIGGER tritbl_sda_irmagr AFTER INSERT
  ON tbl_sda_irm
  FOR EACH ROW
  EXECUTE PROCEDURE funsda_irm_insert();

/*
-----
Creacion de funciones y triggers para RSA
-----
*/

CREATE OR REPLACE FUNCTION funsda_rsa_insert()
  RETURNS trigger AS
$$
BEGIN
  INSERT INTO "tbl_sda_monitoreo" ( "cpnnumerotransmision", "cpccodtransmision", "cpcusuario_servicio",
"cpdfechaultmodificacion", "cpnestadoregistro","cpnid_envioaduana")
  VALUES(NEW."cpnid_rsa",NEW."cpccodtransmision", 'SERV_FACADE', CURRENT_TIMESTAMP, 1,
NEW."cpnid_envioaduana");
  INSERT INTO "tbl_sda_correlativo" ( "cpnnumerotransmision", "cpccodtransmision", "cpnanhocorrelativo")
  VALUES(NEW."cpnid_rsa",NEW."cpccodtransmision", DATE_PART('year', CURRENT_DATE)::INTEGER);
RETURN NEW;
END;
$$
LANGUAGE 'plpgsql';

CREATE TRIGGER tritbl_sda_rsaagr AFTER INSERT
  ON tbl_sda_rsa
  FOR EACH ROW
  EXECUTE PROCEDURE funsda_rsa_insert();

```

Generar diccionario de datos

A continuación se describen los objetos de la base de datos de transmisiones.

TABLAS:

Tabla	TBL_SDA_CONFIGURACION			
Tabla para configurar los códigos, límites, correos, credenciales, auditoria para las transmisiones IRM y RSA				
Campo	Descripción	Tipo dato	PK	FK
cpnid_configuracion	ID secuencial de la tabla	bigserial	Si	
cpccodtransmision	Código de transmisión	char(4)		
cpcsiglatransmision	Sigla abreviada de transmisión	varchar(100)		
cpctipotransmision	Tipo de transmisión	varchar(100)		
cpcdesctransmision	Descripción de transmisión	varchar(100)		
cpctablatransmision	Campos cabecera para monitoreo	varchar(50)		
cpctabladetalletransmision	Campos detalle para monitoreo	varchar(500)		
cpnlimiteminutos	Límite de minutos transcurridos para transmisión	Integer		

cpnumbralalerta	Límite de minutos transcurridos para alerta	Integer		
cpnumbralcritico	Límite de minutos transcurridos para crítico	Integer		
cpccorreotalertausuario	Correo de usuario que recibe la alerta	varchar(500)		
cpnlimreintentoenvio	Límite de minutos transcurridos para reintentar el envío	Integer		
cpnlimreintentoconsulta	Límite de minutos transcurridos para reintentar la consulta	Integer		
cpcusuariows	Usuario de conexión de Servicio Web de SUNAT	varchar(100)		
cpcclavevs	Clave de conexión de Servicio Web de SUNAT	varchar(50)		
cpnestadoconfig	Estado de configuración	samllint		
cpcusuario_servicio	Usuario que registra	varchar(50)		
cpdfechacreacion	Fecha de registro	timestamp		
cpdfechaultmodificacion	Fecha de última modificación	timestamp		
cpccodtransaccionoma	Código de transmisión alternativo	varchar(50)		
cpnlimiteplazoenvio	Límite de plazo de envío	integer		

Tabla	TBL_SDA_CORRELATIVO			
Tabla para los correlativos autogenerados anualmente por cada registro de las transmisiones de IRM y RSA				
Campo	Descripción	Tipo dato	PK	FK
cpnid_correlativo	ID secuencial de la tabla	bigserial	Si	
cpnnumerotransmision	Número de transmisión	bigint		Si
cpccodtransmision	Código de transmisión	char(4)		Si
cpnanhocorrelativo	Año de correlativo	smallint		
cpnseriecorrelativo	Serie de correlativo generado	integer		
cpdfechacreacion	Fecha de creación de registro	timestamp		

Tabla	TBL_SDA_EVENTO			
Tabla que configura los tipos de eventos durante el proceso de transmisión de IRM y RSA				
Campo	Descripción	Tipo dato	PK	FK
cpnid_evento	ID secuencial de la tabla	bigserial	Si	
cpcnombreevento	Nombre de evento. Ejemplos: Pendiente de envío En proceso de transmisión En proceso de consulta Reintentando transmisión Reintentando consulta Desestimado por tiempo Error inesperado de proceso Error de comunicación Envío en procesamiento Transmisión exitosa	varchar(50)		
cpntipoevento	Tipo de evento	smallint		

cpnestadoeventoaduana	Estado evento aduana	smallint		
-----------------------	----------------------	----------	--	--

Tabla	TBL_SDA_CONTROL			
Tabla que registra todos los eventos ejecutados durante todo el proceso de transmisión de IRM y RSA				
Campo	Descripción	Tipo dato	PK	FK
cpnid_control	ID secuencial de la tabla	bigserial	Si	
cpnnumerotransmision	Número de transmisión	bigint		Si
cpccodtransmision	Código de transmisión	char(4)		Si
cpnid_evento	ID de evento	integer		Si
cpncodrespuesta	Código de respuesta del evento	integer		
cpcinfomensajerespuesta	Mensaje de respuesta del evento	text		
cpcrespuestaxml	Mensaje de respuesta del evento en formato xml	text		
cpcusuario_servicio	Usuario auditoria registro de evento	varchar(50)		
cpdfechaevento	Fecha de evento	timestamp		

Tabla	TBL_SDA_MONITOREO			
Tabla con información actualizada del estado, respuestas, errores, logs de las transmisiones de IRM y RSA				
Campo	Descripción	Tipo dato	PK	FK
cpnid_monitoreo	ID secuencial de la tabla	bigserial	Si	
cpnnumerotransmision	Número de transmisión	bigint		Si
cpccodtransmision	Código de transmisión	char(4)		Si
cpnidultimoevento	ID del último evento de la transmisión	integer		
cpdfechaultimoevento	Fecha del último evento de la transmisión	timestamp		
cpcarchivoenvioxml	Documento transmitido en formato xml	text		
cpccadenabase64	Documento transmitido en formato base 64	text		
cpcpeticionenvioxml	Petición completa de envío en formato xml al servicio de SUNAT	text		
cpcacuseenvioxml	Acuse de respuesta de SUNAT en formato xml	text		
cpcanhoenvio	Año de envío de transmisión	char(4)		
cpdfecharecepcionenvio	Fecha de recepción de envío de transmisión	timestamp		
cpcticketenvio	Número de ticket para la consulta de respuesta de la transmisión	char(20)		
cpcpeticionconsultaxml	Petición completa de consulta en formato xml al servicio de SUNAT	text		
cpcrespuestaconsultaxml	Respuesta completa de consulta de SUNAT en formato xml	text		
cpcrespuestaxml	Contenido de respuesta de la consulta en formato ml	text		

cpncodrespuesta	Código de respuesta de la transmisión de SUNAT	integer		
cpcinfomensajerespuesta	Mensaje de respuesta de la transmisión de SUNAT	text		
cpdfcharespuesta	Fecha de respuesta de la transmisión de SUNAT	timestamp		
cpndeshabilitarreintento	Flag de habilitación de reintento	smallint		
cpcusuario_servicio	Usuario auditoria registro	varchar(50)		
cpdfchacreacion	Fecha de registro	timestamp		
cpdfchaultmodificacion	Fecha de última modificación	timestamp		
cpnestadoregistro	Estado de registro	smallint		
cpnid_envioaduanas	ID del envío de aduana	bigint		

Tabla	TBL_SDA_IRM			
Tabla que registra toda la información de los campos de negocio de las transmisiones de IRM				
Campo	Descripción	Tipo dato	PK	FK
cpnid_irm	ID secuencial de la tabla	bigserial	Si	
cpccodtransmision	Código de transmisión	char(4)		Si
cpcmensajjson	Información de los campos de negocio en formato json	json		
cpnidtviajesfecha	ID viaje de ingreso	integer		
cpdfchallegada	Fecha de llegada de mercancía	timestamp		
cpcnumeroviaje	Número de viaje	varchar(200)		
cpcrecalada	Número de recalada	varchar(50)		
cpcanhomanifiesto	Año de manifiesto	char(4)		
cpnnumeromanifiesto	Número de manifiesto	integer		
cpncantidadbultos	Cantidad de bultos de mercancía	integer		
cpnpesobruto	Peso bruto de la mercancía	numeric(13,3)		
cpncantidadbultosmalos	Cantidad de bultos malos de la mercancía	integer		
cpnpesomalos	Peso de bultos malos de la mercancía	numeric(13,3)		
cpdfchaoperacion	Fecha de operación de la transmisión	timestamp		
cpnid_conocimiento	ID conocimiento	bigint		
cpcnrodetalle	Número de detalle	varchar(5)		
cpcnrodocumentotransportemaster	Número de transporte madre de la transmisión	varchar(34)		
cpcnrodocumentotransporte	Número de transporte de la transmisión	varchar(34)		
cpnpasarlcl	Flag transmisión	integer		
cpctipo	Tipo transmisión	varchar(60)		
cpccierrebl	Flag de cierre	varchar(60)		
cpnid_contenedor	ID contenedor	bigint		
cpcnumerocontenedor	Número de contenedor	varchar(60)		
cpnindicasini	Flag de indicador	Integer		

cpnindicasinicontacto	Flag de indicador de contacto	Integer		
cpcnroactasini	Número de catas	varchar(100)		
cpnid_envioaduana	ID envío aduana	bigint		
cpceestadoenvio	Estado envío	varchar(60)		
cpcinformacionmensaje	Información del mensaje	varchar(200)		
cpncantidadbultosmanifestado	Cantidad de bultos en el manifiesto de transmisión	Integer		
cpccodigoaduana	Código de aduana	char(4)		
cpnid_tipoenvioconfigura	ID tipo de envío configura	bigint		
cpnid_condicioncarga	ID condición de carga	bigint		
cpnid_empresasucursal	ID empresa sucursal	bigint		
cpnmintranscurridos	Minutos transcurridos de la transmisión	integer		
cpcemisor	Código de usuario emisor	varchar(50)		
cpctransmisor	Código de usuario transmisor	varchar(50)		
cpcusuarioservicio	Usuario auditoria registro	varchar(50)		
cpdfechacreacion	Fecha de creación de registro	timestamp		
cpdfechaultmodificacion	Fecha de última modificación de registro	timestamp		
Tabla	TBL_SDA_RSA			
Tabla que registra toda la información de los campos de negocio de las transmisiones de RSA				
Campo	Descripción	Tipo dato	PK	FK
cpnid_rsa	ID secuencial de la tabla	bigserial	Si	
cpccodtransmision	Código de transmisión	char(4)		Si
cpcmensajejson	Información de los campos de negocio en formato json	json		
cpnid_salidaalmacen	ID salida almacén	bigint		
cpnid_blmastersalidaalmacen	ID bl master de salida de almacén	bigint		
cpnid_tblmaster	ID master	bigint		
cpnid_tblcontainers	ID contenedor	bigint		
cpnid_tautorizacion	ID autorización	bigint		
cpnid_tticket	ID ticket de salida	bigint		
cpcaduanadeproseso	Código aduana de proceso	varchar(200)		
cpcanniodeclaracion	Año de declaración	varchar(10)		
cpccodigoregimendeclaracion	Código de régimen declaración	varchar(100)		
cpcnumerodeclaracion	Número de declaración	varchar(100)		
cpcplacavehiculo	Placa del vehículo	varchar(50)		
cpcbreveteconductor	Brevete del conductor	varchar(50)		
cpcclasificamercancia	Clasificación de le mercancía	varchar(10)		
cpcidentificaciondeequipamiento	Identificación del equipamiento	varchar(50)		
cpcnumeroprecinto01	Número de precinto	varchar(50)		
cpntotalbulto	Total de bultos	integer		

cpnbultosaldo	Saldo de bultos	numeric(13,3)		
cpnpesobruto	Peso bruto	numeric(13,3)		
cpdfechasalida	Fecha de salida de la mercancía	timestamp		
cpcdescripcionmercancia	Descripción de la mercancía	varchar(200)		
cpnvalidapesobulto	Flag de validación del peso de bulto	Integer		
cpnsalidaparcial	Flag de salida parcial del bulto	Integer		
cpncodigoestadoenvio	Código de estado de envío	integer		
cpcnumerocorrelativodeclaracion	Número de correlativo de la declaración	varchar(100)		
cpcnumautorizacion	Número de autorización	varchar(100)		
cpcnumdocuadua	Número de documento aduana	varchar(100)		
cpcobservacion	Observación de la transmisión	varchar(200)		
cpcdescripcionrespuesta	Descripción de respuesta de la transmisión	varchar(200)		
cpnid_envioaduana	ID envío aduana	bigint		
cpnid_empresasucursal	ID empresa sucursal	bigint		
cpnid_tipoenvioconfigura	ID tipo de envío de configuración	bigint		
cpnid_condicioncarga	ID condición de carga	bigint		
cpnmintranscurridos	Minutos transcurridos de la transmisión	integer		
cpcrecalada	Número de recalada	varchar		
cpcnumbld	Número de bl	varchar		
cpnnumerovolante	Número de volante	integer		
cpcnumerodam	Número dam	varchar		
cpcrucrtransportista	RUC transportista	varchar		
cpntaracamion	Peso tara camión	numeric(13,3)		
cpnpesoneto	Peso neto	numeric(13,3)		
cpcmanifiesto	Número de manifiesto	varchar		
cpcnumdetalle	Número de detalle	varchar		
cpcusuarioservicio	Usuario auditoria registro	varchar(50)		
cpdfechacreacion	Fecha de creación de registro	timestamp		
cpdfechaultmodificacion	Fecha de última modificación de registro	timestamp		

Tabla	TBL_SDA_REINTENTO			
Tabla que registra todos los eventos de reintentos de transmisión de IRM y RSA				
Campo	Descripción	Tipo dato	PK	FK
cpnid_reintento	ID secuencial de la tabla	bigserial	Si	
cpnnumerotransmision	Número de transmisión	bigint		Si
cpcodtransmision	Código de transmisión	char(4)		Si
cpdfechareintento	Fecha de reintento de transmisión	timestamp		

cpctiporeintento	Tipo de reintento SERV_REINTENTO_TRANSMISION SERV_REINTENTO_CONSULTA	varchar(50)		
cpcpeticionreintentoxml	Documento de petición en formato xml	text		

Tabla	TBL_SDA_REINTENTO_CONFIGURACION			
Tabla que registra la configuración de las transmisiones pendientes de IRM y RSA				
Campo	Descripción	Tipo dato	PK	FK
cpnid_reintentoconf	ID secuencial de la tabla	bigserial	Si	
cpccodtransmision	Código de transmisión	char(4)		Si
cpnnumeroreintento	Fecha de reintento de transmisión	integer		
cpnminutosreintentoenvio	Minutos transcurridos para el reintento de envío	integer		
cpnminutosreintentoconsulta	Minutos transcurridos para el reintento de consulta	integer		
cpdfechacreacion	Fecha de creación de registro	timestamp		
cpdfechaultmodificacion	Fecha de última modificación de registro	timestamp		

Tabla	TBL_SDA_REINTENTO_ENVIO			
Tabla que registra los reintentos de transmisiones pendientes de envío de IRM y RSA				
Campo	Descripción	Tipo dato	PK	FK
cpnid_reintentoenvio	ID secuencial de la tabla	bigserial	Si	
numero_transmision	Número de transmisión	varchar(20)		Si
codigo_transmision	Código de transmisión	varchar(4)		
informacion_transmision	Información a enviar en el reintento	varchar(4000)		
nro_reintento	Número de reintento	varchar(5)		
cpceestadoenvio	Estado de reintento True = enviado False = pendiente	varchar(4)		
cpdfechacreacion	Fecha de creación de registro	timestamp		
cpdfechaultmodificacion	Fecha de última modificación de registro	timestamp		

Tabla	TBL_SDA_REINTENTO_CONSULTA			
Tabla que registra los reintentos de transmisiones pendientes de consulta de IRM y RSA				
Campo	Descripción	Tipo dato	PK	FK
cpnid_reintentoconsulta	ID secuencial de la tabla	bigserial	Si	
numero_transmision	Número de transmisión	varchar(20)		Si
codigo_transmision	Código de transmisión	varchar(4)		
ticket_envio	Ticket para hacer el reintento de consulta	varchar(10)		
anho_envio	Año de ticket para hacer el reintento de consulta	char(4)		
nro_reintento	Número de reintento	varchar(5)		

cpceestadoenvio	Estado de reintento True = enviado False = pendiente	varchar(10)		
cpdfechacreacion	Fecha de creación de registro	timestamp		
cpdfechaultmodificacion	Fecha de última modificación de registro	timestamp		

FUNCIONES:

Función	FUN_SDA_IRM_INSERT
Función que insertan nuevos registros en las TBL_SDA_MONITOREO y TBL_SDA_CORRELATIVO para las transmisiones IRM	
Descripción de lógica	
1.- Insertar registro en la tabla TBL_SDA_MONITOREO para el monitoreo de la transmisión IRM. 2.- Insertar registro en la tabla TBL_SDA_CORRELATIVO con el correlativo de la transmisión IRM.	

Función	FUN_SDA_RSA_INSERT
Función que insertan nuevos registros en las TBL_SDA_MONITOREO y TBL_SDA_CORRELATIVO para las transmisiones RSA	
Descripción de lógica	
1.- Insertar registro en la tabla TBL_SDA_MONITOREO para el monitoreo de la transmisión RSA. 2.- Insertar registro en la tabla TBL_SDA_CORRELATIVO con el correlativo de la transmisión RSA.	

TRIGGERS:

Trigger	TRITBL_SDA_IRMAGR
Trigger para los eventos de inserción de registros IRM	
Descripción de lógica	
1.- Al insertar un nuevo registro en IRM, llamar a la función FUN_SDA_IRM_INSERT	

Trigger	TRITBL_SDA_RSAAGR
Trigger para los eventos de inserción de registros RSA	
Descripción de lógica	
1.- Al insertar un nuevo registro en RSA, llamar a la función FUN_SDA_RSA_INSERT	

g.- Sprint 2.

Como primera actividad se programó la reunión de Sprint Planning con los miembros del equipo básico scrum. El Scrum master y el Equipo Scrum definieron las tareas del Sprint backlog para el Sprint 2.

HISTORIAS DE USUARIO (HU)	TO DO	RECURSOS
E.3 HU.1 Diseñar componentes de Arquitectura	E.3 HU.1 T.1 Diseñar Arquitectura de Integración de solución	Arquitecto de Software
	E.3 HU.1 T.2 Diseñar Flujos de procesos de componentes de integración	Arquitecto de Software
	E.3 HU.1 T.3 Diseñar solución de escalabilidad de reintentos	Arquitecto de Software
E.3 HU.2 Desarrollar componentes de Arquitectura	E.2 HU.2 T.1 Desarrollar framework de integración con wso2	Arquitecto de Software
	E.2 HU.2 T.2 Desarrollar componentes de integración ESB con wso2	Arquitecto de Software

Figura 11-Sprint Backlog del Sprint 02 (Fuente: Elaboración Propia)

Ejecución de tareas.

Diseñar Arquitectura de Integración de solución

El Arquitecto de Software diseñó un diagrama de solución tecnológica que integra los componentes tecnológicos desde las propias instalaciones de TPP, la nueva arquitectura de integración en aws y los componentes web services de SUNAT.

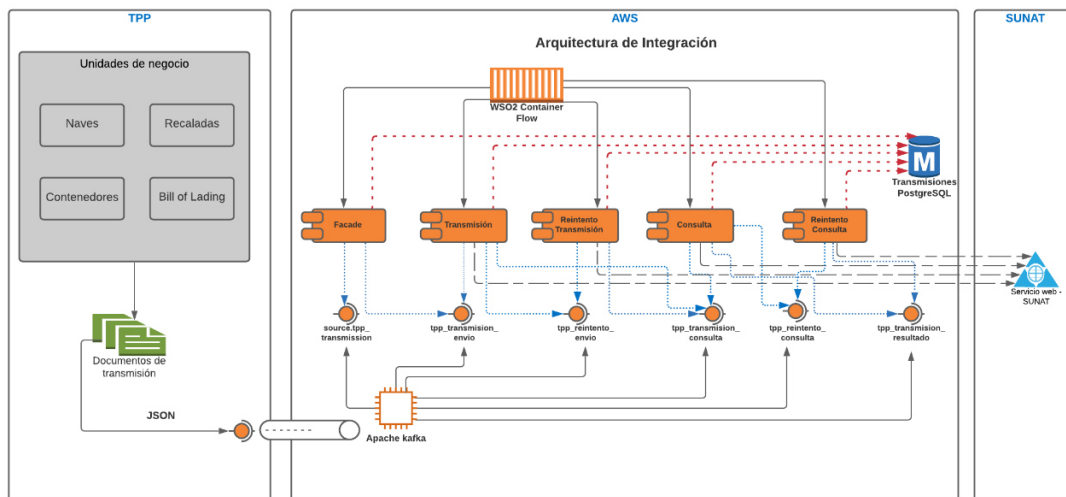


Figura 12-Diagrama de Solución Tecnológica (Fuente: Elaboración Propia)

Se describe los componentes de la solución tecnológica:

- Componentes tecnológicos de TPP.

Documentos de transmisión: Son objetos de información en formato JSON que contienen los campos de negocios analizados en el mapeo del Sprint 1 para IRM y RSA. Estos documentos en formato JSON serán enviados desde las aplicaciones “onpremise” de TPP a las colas de entradas de mensajes (kafka) en nube de la arquitectura de integración. Es responsabilidad de las unidades de negocio de TPP analizadas en el alcance organizacional enviar esta información.

- Arquitectura de integración.

Los componentes software de la nueva arquitectura de integración en aws a implementar son las siguientes:

Kafka nube: Gestor de colas que publica las colas de entradas de mensajes que son enviados desde las unidades de negocio de TPP, las siguientes son las colas configuradas:

- source.tpp_transmision: Cola de entrada de mensajes que reciben los documentos a transmitir en formato JSON desde las unidades de negocios de TPP. Son procesados por el componente Facade.
- tpp_transmision_envio: Cola de entrada de mensajes que reciben los documentos a transmitir en formato JSON desde el componente Facade. Son procesados por el componente Transmisiones.
- tpp_reintento_envio: Cola de entrada de mensajes que reciben los documentos a retransmitir en formato JSON desde el componente Transmisiones y Reintento de Transmisiones. Son procesados por el componente Reintento Transmisiones.
- tpp_transmision_consulta: Cola de entrada de mensajes que reciben los números y códigos de transmisión en formato JSON desde el componente Transmisiones para la consulta de estados. Son procesados por el componente Consulta.
- tpp_reintento_consulta: Cola de entrada de mensajes que reciben los números y códigos de transmisión en formato JSON desde el componente Consulta y Reintento de Consulta para la consulta de estados. Son procesados por el componente Reintento de Consulta.
- tpp_transmision_resultado: Cola de entrada de mensajes que reciben las respuestas de la consulta de estados de transmisiones en formato JSON desde el componente Consulta o Reintento de Consulta.

Facade: Componente software de integración que realiza la validación y transformación de los campos de negocios del documento de transmisión “precisar los procesos, fuente y destino”.

Transmisiones: Componente software de integración que realiza la generación y envío del documento de transmisión en formato cuscar-xml. En caso de éxito envía la petición a la cola de componente de software de consulta, en caso de error durante el proceso envía la petición a la cola del componente software de reintento de transmisión.

Reintento Transmisión: Componente software de integración que realiza el reintento de envío del documento de transmisión.

Consulta: Componente software de integración que realiza la consulta de los estados de transmisión.

Reintento Consulta: Componente software de integración que realiza el reintento de consulta del estado de la transmisión.

WSO2 Container Flow: Contenedor ESB de permite desplegar e integrar los componentes de la arquitectura.

Transmisiones PostgreSQL: Base de Datos que soporta las configuraciones, transacciones de las transmisiones de documentos a SUNAT.

- SUNAT.

Servicios Web - SUNAT: Servicios Web de SUNAT para la recepción y consulta de las transmisiones.

Diseñar Flujos de procesos de componentes de integración

Flujo de Facade:

El primer flujo recibe la información enviadas por las unidades de negocios en formato JSON. Como primer paso valida que los campos obligatorios no estén vacíos o tengan formato de tipo de dato incorrecto, en estos casos notifica al operador de la unidad a través de un correo electrónico.

Como segundo paso transforma los campos al formato solicitado por la plantilla de SUNAT, en caso de error durante la transformación también notifica al operador de la unidad a través de un correo.

La información validada y transformada es enviada a la cola de ingreso del componente de software de Transmisión.

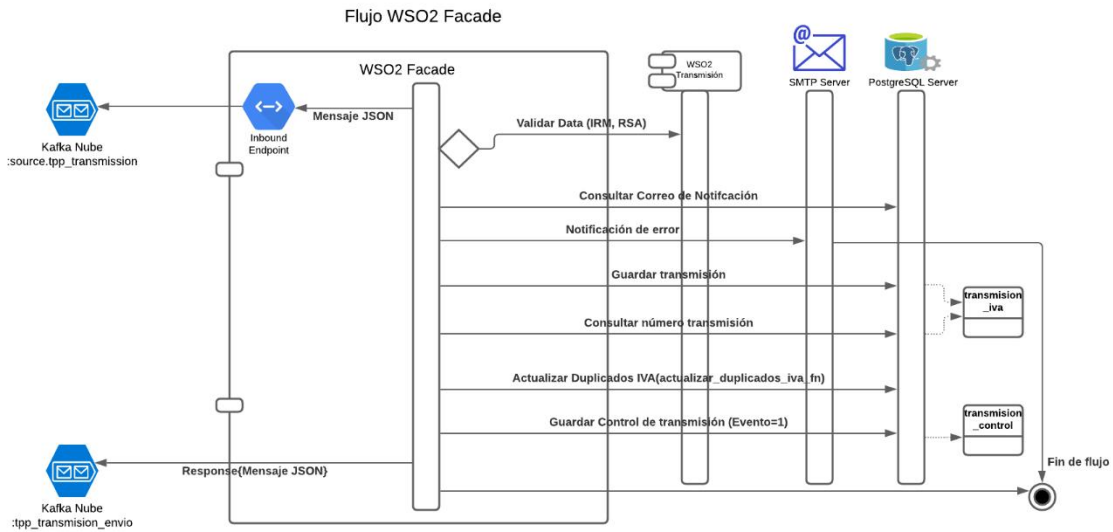


Figura 13-Diagrama de Flujo del componente Facade (Fuente: Elaboración Propia)

Flujo de Transmisión:

El proceso de transmisión inicia con la recepción de la información validada y transformada por Facade. Con esta información el componente construye el documento a transmitir en formato requerido por SUNAT (cuscar-xml). Posteriormente envía dicho documento a SUNAT (Web Services) según sea la operación requerida (Ingreso Recepción de Mercancías (IRM), Registro de salida de Mercancías (RSA)). La SUNAT responde con un acuse de recibo y el número de operación que serán usados en el componente de software de Consulta.

En caso exista un error en el proceso de Transmisión la operación será enviada a la cola de ingreso del componente de software Reintento de Transmisiones.

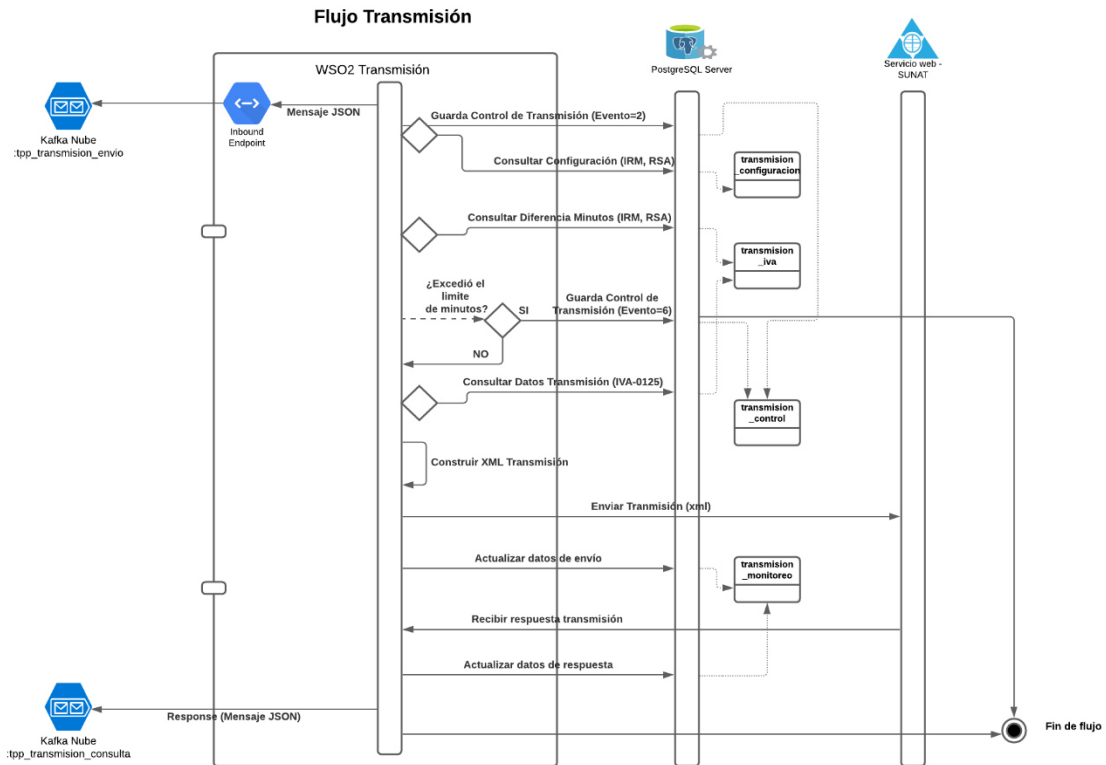


Figura 14-Diagrama de Flujo del componente Transmisión (Fuente: Elaboración Propia)

Flujo de Consulta

El proceso de Consulta inicia con la recepción de la información de acuse de recibo y el número de operación de SUNAT. Se genera una petición a SUNAT (Web Service) para verificar el estado de la transmisión (Transmisión exitosa, Error de validación, Sin respuesta de SUNAT). Mientras la operación no sea exitosa enviará la información de acuse de recibo y el número de operación (Con un límite configurado) a la cola de ingreso del componente Software de Reintento de Consulta. Si existe una transmisión exitosa el estado será enviado a la cola de ingreso de Resultados de Transmisión.

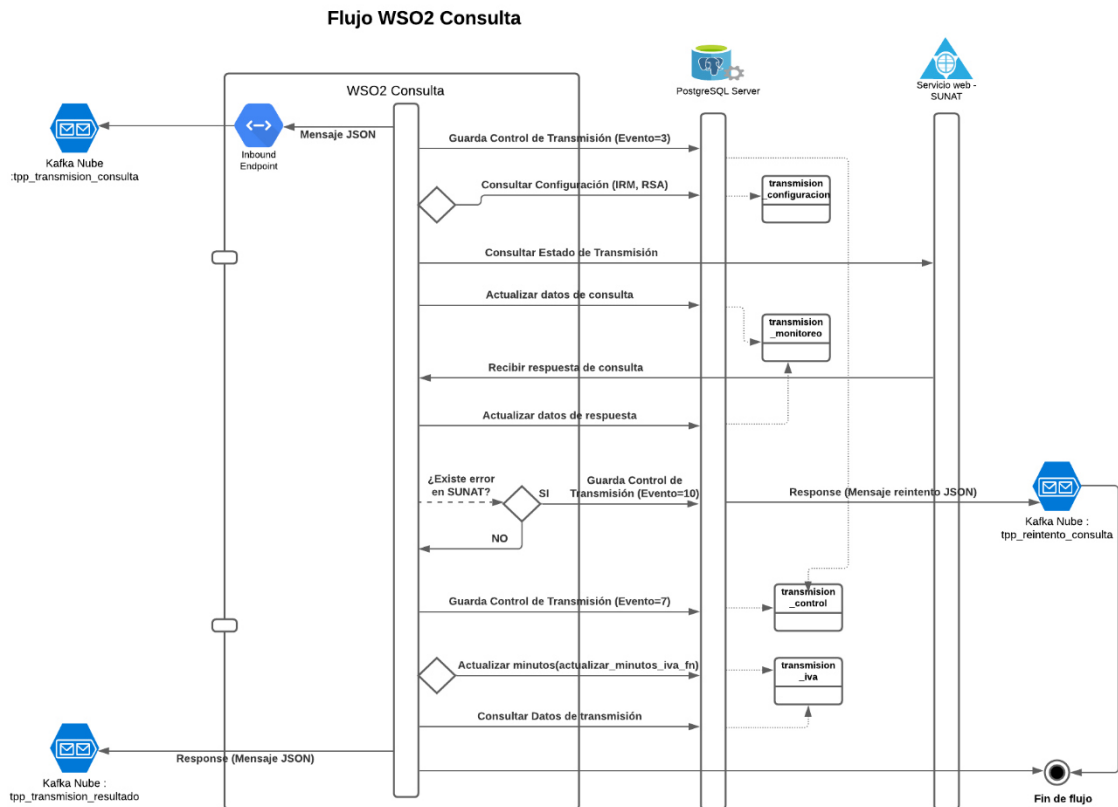


Figura 15-Diagrama de Flujo del componente Consulta (Fuente: Elaboración Propia)

Flujo de Reintento de Transmisiones:

El proceso de Reintento de Transmisión inicia con la recepción de la información del número de operación y código de operación (IRM, RSA) enviados por el componente de Transmisión. Con esta información el componente consulta desde las tablas de la Base de Datos de Transmisiones el documento a transmitir en formato requerido por SUNAT (cuscar-xml). Nuevamente envía dicho documento a SUNAT (Web Services) según sea la operación requerida (Ingreso Recepción de Mercancías (IRM), Registro de salida de Mercancías (RSA)). La SUNAT responde con un acuse de recibo y el número de operación que serán usados en el componente de software de Consulta.

Si persisten los errores en el proceso reenvía a la cola de ingreso del componente de Reintento de Transmisiones con límite de 3 reintentos cada 5 segundos. Desde el cuarto reintento el componente registra en una tabla de reintentos pendientes de transmisión y lo procesa cada 5 min hasta el límite máximo de tiempo a transmitir (IRM límite de 30 minutos desde el ingreso a al flujo, RSA límite de 24 horas desde el ingreso al flujo).

Flujo WSO2 Reintento Transmisiones

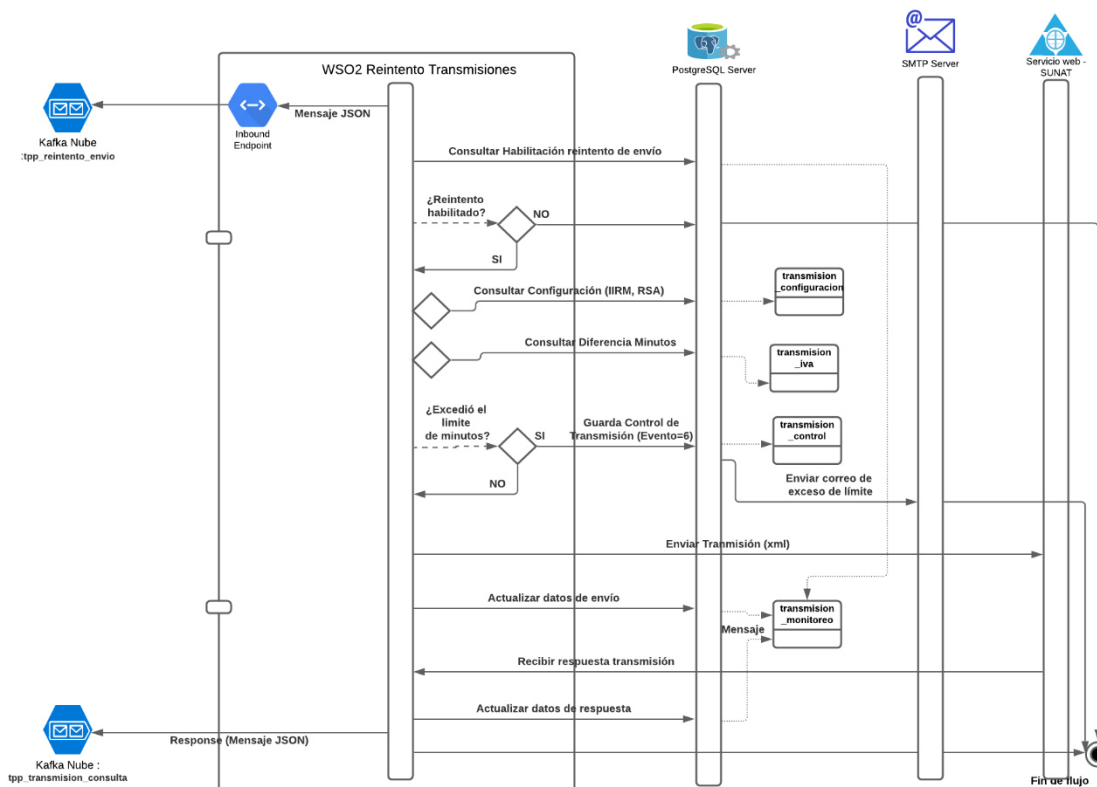


Figura 16-Diagrama de Flujo del componente Reintento de Transmisiones (Fuente: Elaboración Propia)

Flujo de Reintento de Consulta:

El proceso de Reintento de Consulta inicia con la recepción de la información del acuse de recibo y número de operación enviados por el componente de Consulta. Nuevamente realiza la consulta del estado de transmisión a SUNAT (Web Services) según sea la operación requerida (Ingreso Recepción de Mercancías (IRRM), Registro de salida de Mercancías (RSA)).

Si existe una transmisión exitosa el estado será enviado a la cola de ingreso de Resultados de Transmisión.

Si persisten los errores en el proceso reenvía a la cola de ingreso del componente de Reintento de Consulta con límite de 3 reintentos cada 5 segundos. Desde el cuarto reintento el componente registra en una tabla de reintentos pendientes de consulta y lo procesa cada 5 min hasta el límite máximo de tiempo a transmitir (IRRM límite de 30 minutos desde el ingreso a al flujo, RSA límite de 24 horas desde el ingreso al flujo).

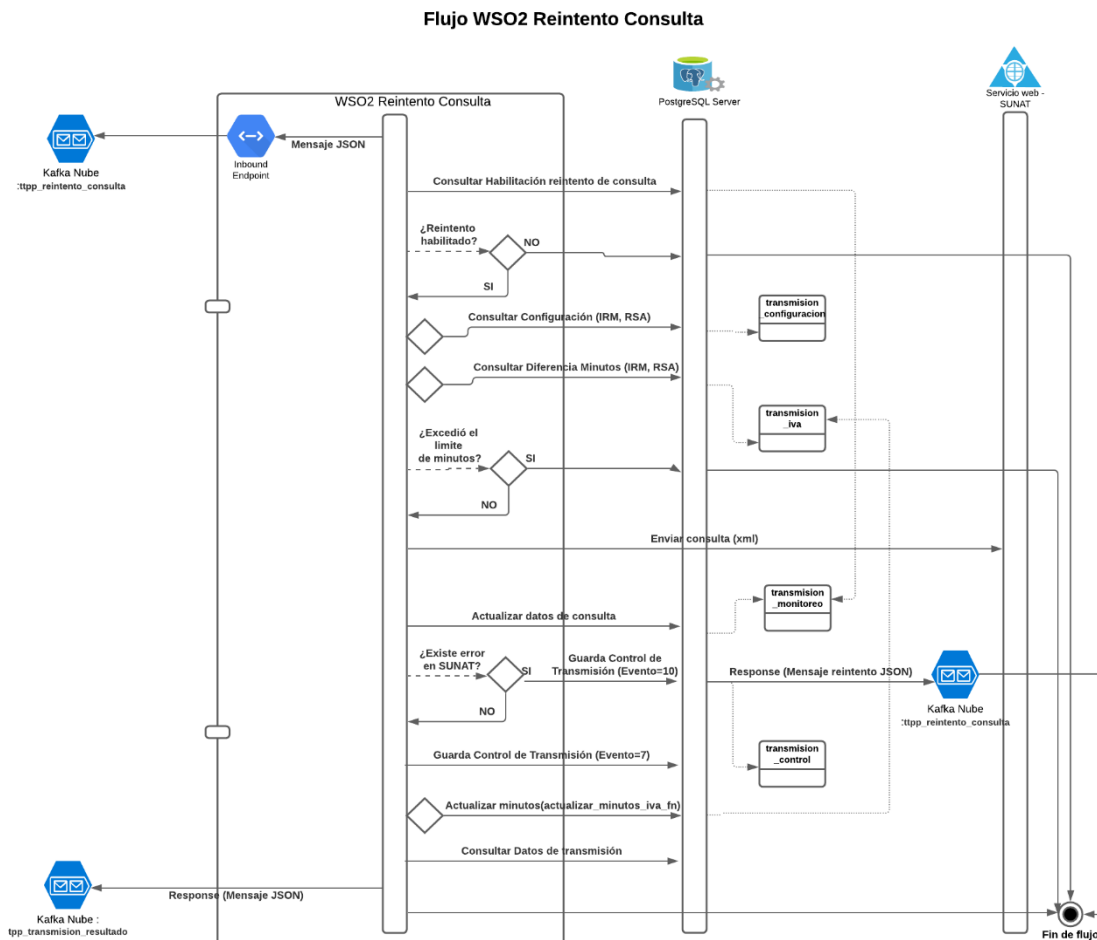


Figura 17-Diagrama de Flujo del componente Reintento de Consulta (Fuente: Elaboración Propia)

Diseñar solución de escalabilidad de reintentos

La solución técnica se divide en dos partes:

- Solución de escalabilidad de reintentos de transmisiones: En el proceso del flujo de Transmisión se pueden producir errores de comunicación y timeout con SUNAT, cuando esto sucede se envía la transmisión al Flujo de Reintento de Transmisión. Los números de reintentos no pueden ser ilimitados y debe evitar que se junten a otras retransmisiones, para ello la solución permite hacer 3 reintentos continuos (Cada 5 segundos) usando la cola de mensaje de entrada del componente de Reintento de Transmisiones, a partir del cuarto reintento serán registrados en una tabla de pendientes de reintento de transmisión (tbl_sda_reintento_envio) para que la petición se haga en incrementos de cada 5 min hasta un límite de tiempo según configuración (30 min). El reintento incremental de transmisiones y con un límite soluciona problemas de escalabilidad en la arquitectura de integración.

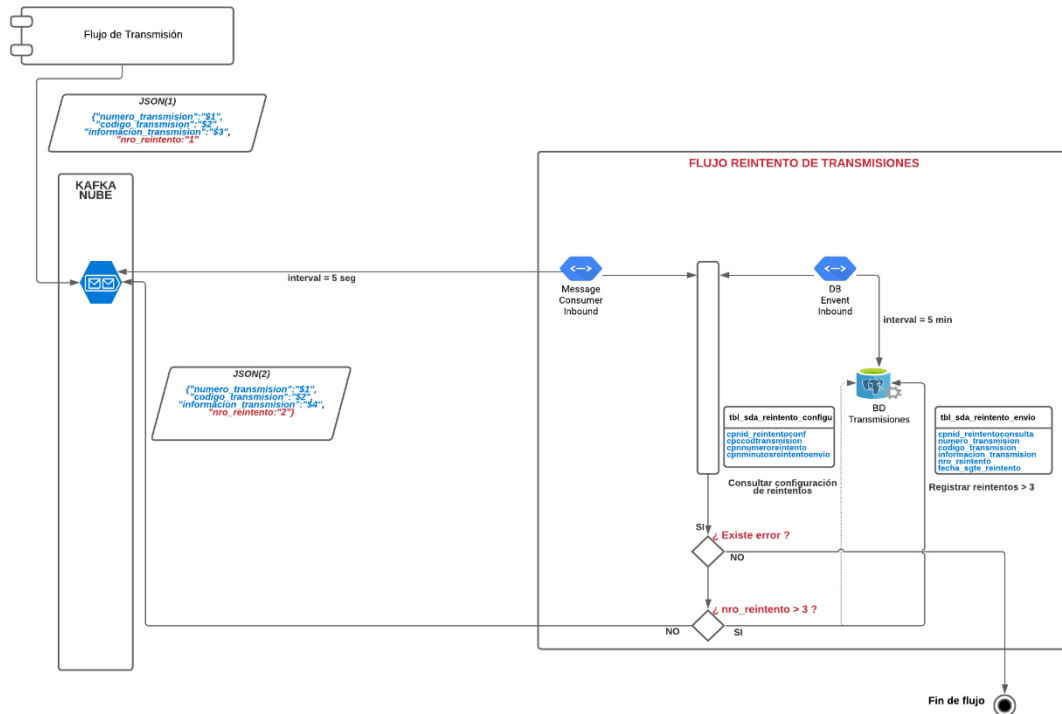


Figura 18-Diagrama de solución de escalabilidad de Reintento de Transmisiones (Fuente: Elaboración Propia)

- Solución de escalabilidad de reintentos de consulta: La solución tiene el mismo esquema que el anterior, solo se modifica que es el proceso de flujo de consulta quien produce el error y se envía al Flujo de reintento de consulta, así como la tabla de pendientes de reintento de consulta (tbl_sda_reintento_consulta) para que las peticiones. El reintento incremental de consulta y con un límite soluciona problemas de escalabilidad en la arquitectura de integración.

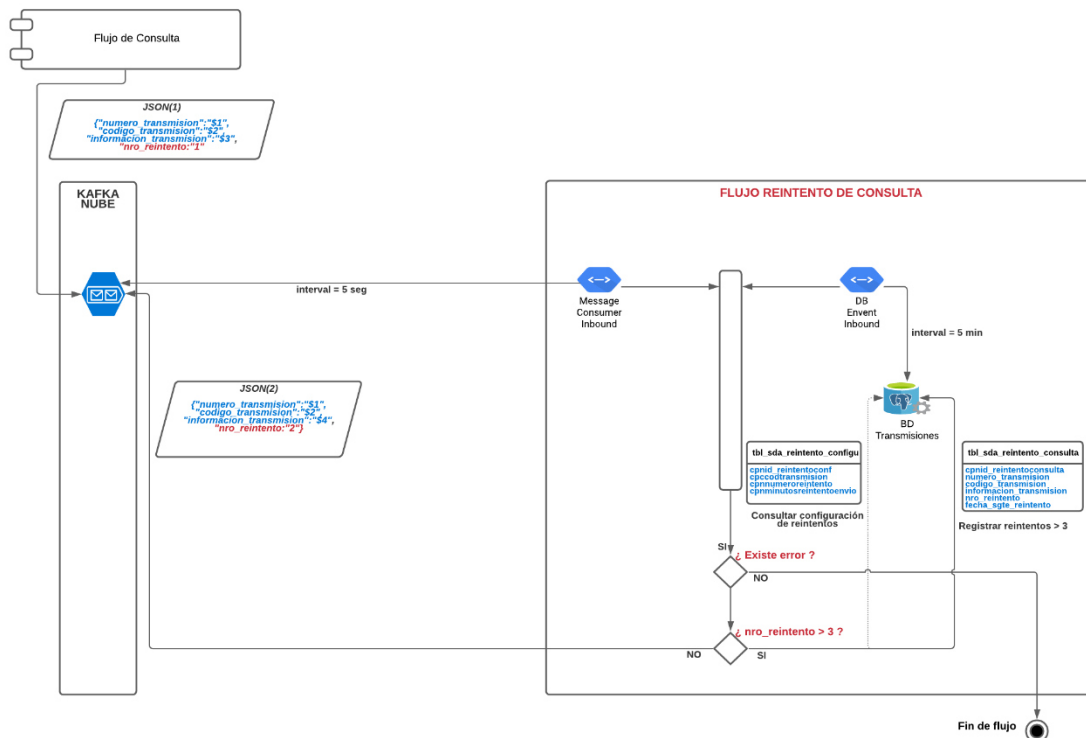


Figura 19-Diagrama de solución de escalabilidad de Reintento de Consulta (Fuente: Elaboración Propia)

Desarrollar framework de integración con wso2

Una de las ventajas de usar el Integrador empresarial wso2 es que ya tienen artefactos predefinidos, plantillas de componentes para el desarrollo de la estructura de la solución. Describimos los usados para el desarrollo de los componentes:

- Inbound Endopints: Artefacto que permite desarrollar entradas de eventos para el inicio de un flujo de proceso. Eventos que están leyendo información periódicamente de una fuente.
- Sequence: Artefacto que permite desarrollar la lógica de los pasos de los flujos de procesos.w
- Drop: Artefacto para finalizar un flujo de procesos.
- Log: Artefacto que permite desarrollar mensajes de auditoria en los flujos de procesos.
- Connection: Artefacto para crear una conexión a Base de datos.
- Statement: Artefacto para desarrollar sentencias SQL.
- PayloadFactory: Artefacto para desarrollar estructuras de objetos en formato JSON y XML.
- kafkaTransport: Artefacto para desarrollar una conexión y envío de mensajes a una cola Kafka.
- LocalEntry: Artefacto para desarrollar una plantilla de intercambio de datos (cuscar-xml) solicitado por SUNAT.

- Class: Artefacto para desarrollar clases con lógica personalizada dentro de un flujo de proceso.
- Call: Artefacto para desarrollar clientes Web Services.
- Property: Artefacto para desarrollar variables dentro de los flujos de procesos.

En base los artefactos predefinidos se define las estructuras de los proyectos (framework):

-Estructura del Proyecto del Componente Facade.

Se describe los objetos del componente.

1.- FacadeCustomWSO2Mediators: Subcomponente que contiene los paquetes y clases para la notificación de alertas por correo, lógica de validación y transformación de datos de IRM y RSA.

Los objetos implementados en código Java para Facade son los siguientes:

pe/com/tpp/mediators/facade/ProcesarDataIRM.java

pe/com/tpp/mediators/facade/ProcesarDataRSA.java

pe/com/tpp/mediators/facade/EnviarAlerta.java

1.2.-lib: Carpeta de librerías de externas de Facade.

1.3.- pom.xml: Archivo Maven para la gestión, compilación y empaquetado del subcomponente.

2.- FacadeWSO2ServiceCompositeExporter: Subcomponente que permite empaquetar todos los subcomponentes de Facade en extensión comprimida (CAR).

3.- FacadeWSO2ServiceConfigs: Subcomponente que permite desarrollar las entradas de eventos (Inbound Endpoints) y los flujos de procesos diseñados (Sequences).

Los objetos implementados para Facade son los siguientes:

src/main/synapse-config/inbound-endpoints/FacadeInboundEP.xml

src/main/synapse-config/sequences/FacadeMainSeq.xml

src/main/synapse-config/sequences/FacadeMainErrorSeq.xml

4.- FacadeWSO2ServiceConnectorExporter: Subcomponente que permite la configuración de librerías para el consumo de mensajes de las colas en Kafka.

5.- FacadeWSO2ServiceDockerExporter: Subcomponente que permite generar una imagen en Docker para el despliegue del componente Facade en un entorno en nube.

-Estructura del Proyecto del Componente Transmisión.

Se describe los objetos del componente.

1.- CustomWSO2Mediators: Subcomponente que contiene los paquetes y clases para la generación del documento de transmisión, creación del cliente de comunicación el envío a los Web Services de IRM y RSA de SUNAT. Los objetos implementados en código Java para Transmisión son los siguientes:

```
pe/com/tpp/mediators/transmisiones/BuildIRMXML.java  
pe/com/tpp/mediators/transmisiones/BuildRSAXML.java  
pe/com/tpp/mediators/transmisiones/TransformRSAResponse.java
```

1.2.-lib: Carpeta de librerías de externas de Transmisión.

1.3.- pom.xml: Archivo Maven para la gestión, compilación y empaquetado del subcomponente.

2.- ConsultaWSO2ServiceCompositeExporter: Subcomponente que permite empaquetar todos los subcomponentes de Transmisión en extensión comprimida (CAR).

3.- TransmisionWSO2ServiceConfigs: Subcomponente que permite desarrollar las entradas de eventos (Inbound Endpoints), los flujos de procesos diseñados (Sequences) y las plantillas de intercambio de datos (cuscar-xml) para IRM y RSA.

Los objetos implementados para Transmisión son los siguientes:

```
src/main/synapse-config/inbound-endpoints/TransmisionInboundEP.xml  
src/main/synapse-config/local-entries/LE_TRANSMISION_IRM.xml  
src/main/synapse-config/local-entries/LE_TRANSMISION_IVA.xml  
src/main/synapse-config/sequences/TransmisionMainSeq.xml  
src/main/synapse-config/sequences/TransmisionMainErrorSeq.xml  
src/main/synapse-config/sequences/TransmisionSunatIRMSeq.xml  
src/main/synapse-config/sequences/TransmisionSunatIRMErrorSeq.xml  
src/main/synapse-config/sequences/TransmisionSunatRSASeq.xml  
src/main/synapse-config/sequences/TransmisionSunatRSAErrorSeq.xml
```

4.- TransmisionWSO2ServiceConnectorExporter: Subcomponente que permite la configuración de librerías para el consumo de mensajes de las colas en Kafka.

5.- TransmisionWSO2ServiceDockerExporter: Subcomponente que permite generar una imagen en Docker para el despliegue del componente Transmisión en un entorno en nube.

6.- TransmisionWSO2ServiceRegistryResources: Subcomponente que permite la configuración de recursos properties.

-Estructura del Proyecto del Componente Consulta.

Se describe los objetos del componente.

1.- ConsultaCustomWSO2Mediators: Subcomponente que contiene los paquetes y clases para la creación del cliente de comunicación de consulta al Web Services de consulta. Los objetos implementados en código Java para Consulta son los siguientes:

pe/com/tp/mediators/consulta/TransformResponse.java

1.2.-lib: Carpeta de librerías de externas de Consulta.

1.3.- pom.xml: Archivo Maven para la gestión, compilación y empaquetado del subcomponente.

2.- ConsultaWSO2ServiceCompositeExporter: Subcomponente que permite empaquetar todos los subcomponentes de Consulta en extensión comprimida (CAR).

3.- ConsultaWSO2ServiceConfigs: Subcomponente que permite desarrollar las entradas de eventos (Inbound Endpoints), los flujos de procesos diseñados (Sequences).

Los objetos implementados para Consulta son los siguientes:

src/main/synapse-config/inbound-endpoints/ConsultaInboundEP.xml

src/main/synapse-config/sequences/ConsultaMainSeq.xml

src/main/synapse-config/sequences/ConsultaMainErrorSeq.xml

src/main/synapse-config/sequences/ConsultaSunatSeq.xml

src/main/synapse-config/sequences/ConsultaSunatErrorSeq.xml

4.- ConsultaWSO2ServiceConnectorExporter: Subcomponente que permite la configuración de librerías para el consumo de mensajes de las colas en Kafka.

5.- ConsultaWSO2ServiceDockerExporter: Subcomponente que permite generar una imagen en Docker para el despliegue del componente Consulta en un entorno en nube.

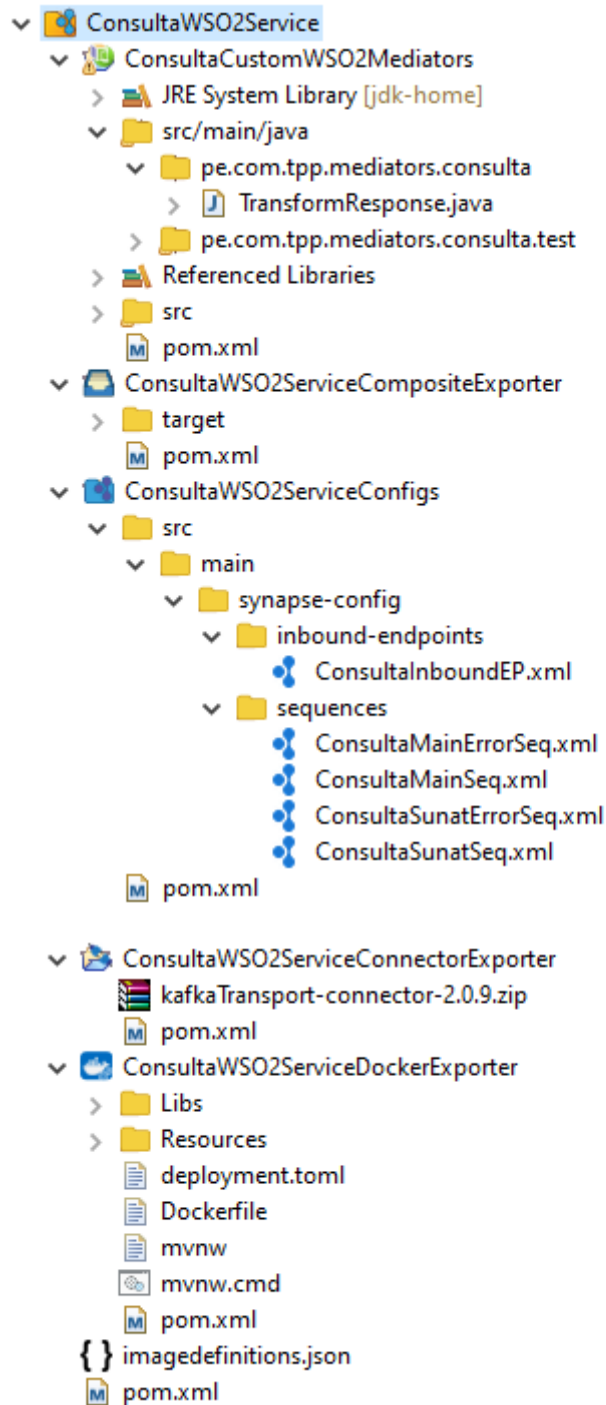


Figura 22-Estructura de proyecto del componente Consulta (Fuente: Elaboración Propia)

-Estructura del Proyecto del Componente Reintento Transmisión.

Se describe los objetos del componente.

- 1.- ReintentoTransmisionCustomWSO2Mediators: Subcomponente que contiene los paquetes y clases para el reintento de envío de transmisión, creación del cliente de comunicación de envío

a los Web Services de SUNAT y la notificación de límite de reintentos. Los objetos implementados en código Java para Reintento Transmisión son los siguientes:

```
pe/com/tpp/mediators/reintentotransmision/EnviarAlerta.java
pe/com/tpp/mediators/reintentotransmision/TransformResponse.java
pe/com/tpp/mediators/reintentotransmision/ValidarReintento.java
pe/com/tpp/mediators/reintentotransmision/ValidarReintentoFecha.java
```

1.2.-lib: Carpeta de librerías de externas de Reintento Transmisión.

1.3.- pom.xml: Archivo Maven para la gestión, compilación y empaquetado del subcomponente.

2.- ReintentoTransmisionWSO2ServiceCompositeExporter: Subcomponente que permite empaquetar todos los subcomponentes de Reintento Transmisión en extensión comprimida (CAR).

3.- ReintentoTransmisionWSO2ServiceConfigs: Subcomponente que permite desarrollar las entradas de eventos (Inbound Endpoints), los flujos de procesos diseñados (Sequences).

Los objetos implementados para Reintento Transmisión son los siguientes:

```
src/main/synapse-config/inbound-endpoints/ReintentoEnvioDBEventInboundEP.xml
src/main/synapse-config/inbound-endpoints/ReintentoEnvioInboundEP.xml
src/main/synapse-config/sequences/ReintentoEnvioMainErrorSeq.xml
src/main/synapse-config/sequences/ReintentoEnvioMainSeq.xml
src/main/synapse-config/sequences/ReintentoEnvioSunatIRMErrorSeq.xml
src/main/synapse-config/sequences/ReintentoEnvioSunatIRMSeq.xml
src/main/synapse-config/sequences/ReintentoEnvioSunatRSAErrorSeq.xml
src/main/synapse-config/sequences/ReintentoEnvioSunatRSASeq.xml
```

4.- ReintentoTransmisionWSO2ServiceConnectorExporter: Subcomponente que permite la configuración de librerías para el consumo de mensajes de las colas en Kafka.

5.- ReintentoTransmisionWSO2ServiceDockerExporter: Subcomponente que permite generar una imagen en Docker para el despliegue del componente Reintento Transmisión en un entorno en nube.

6.- ReintentoTransmisionWSO2ServiceRegistryResources: Subcomponente que permite la configuración de recursos properties.

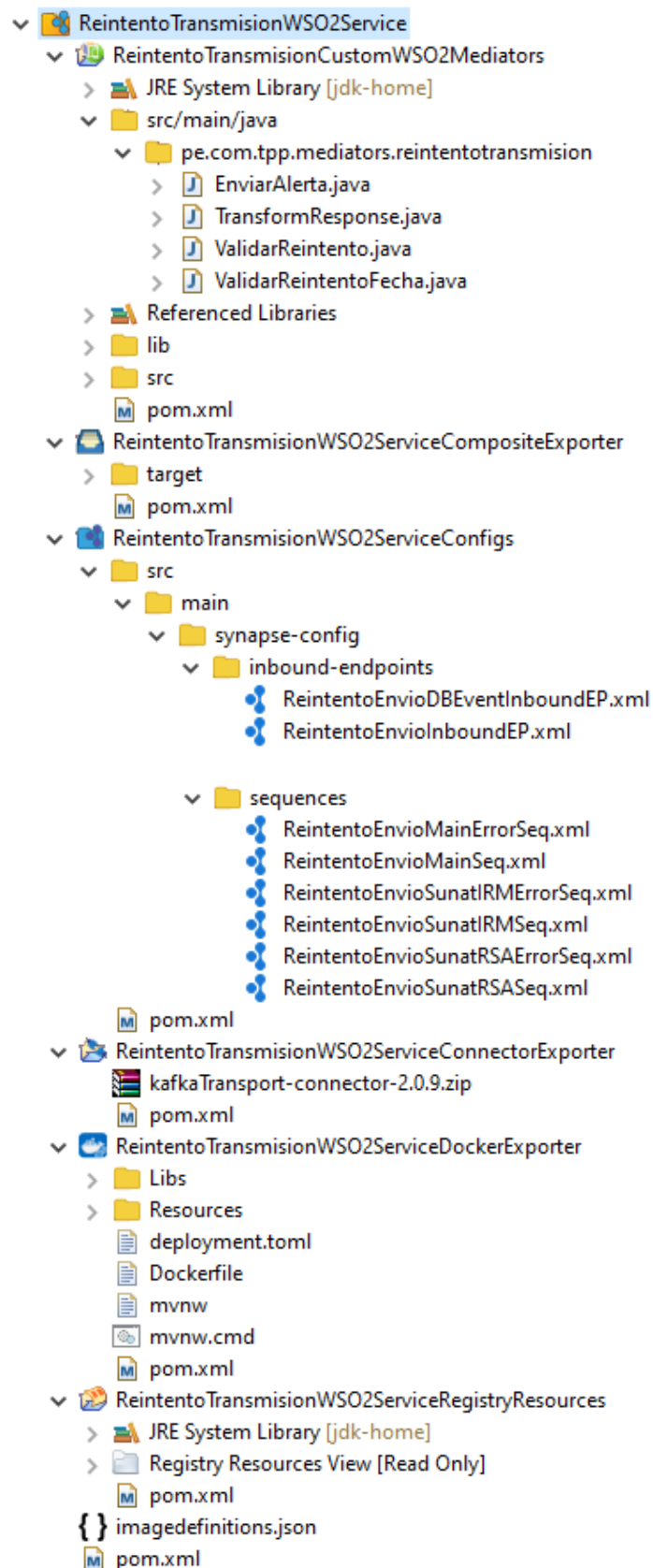


Figura 23-Estructura de proyecto del componente Reintento Transmisión (Fuente: Elaboración Propia)

-Estructura del Proyecto del Componente Reintento Consulta.

Se describe los objetos del componente.

1.- ReintentoConsultaCustomWSO2Mediators: Subcomponente que contiene los paquetes y clases para el reintento de consulta de estado de transmisiones, creación del cliente de comunicación de consulta a los Web Services de SUNAT y la notificación de límite de reintentos. Los objetos implementados en código Java para Reintento Consulta son los siguientes:

```
pe/com/tpp/mediators/reintentoconsulta/TransformResponse.java  
pe/com/tpp/mediators/reintentoconsulta/ValidarReintento.java  
pe/com/tpp/mediators/reintentoconsulta/ValidarReintentoFecha.java
```

1.2.-lib: Carpeta de librerías de externas de Reintento Transmisión.

1.3.- pom.xml: Archivo Maven para la gestión, compilación y empaquetado del subcomponente.

2.- ReintentoConsultaWSO2ServiceCompositeExporter: Subcomponente que permite empaquetar todos los subcomponentes de Reintento Consulta en extensión comprimida (CAR).

3.- ReintentoConsultaWSO2ServiceConfigs: Subcomponente que permite desarrollar las entradas de eventos (Inbound Endpoints), los flujos de procesos diseñados (Sequences).

Los objetos implementados para Reintento Consulta son los siguientes:

```
src/main/synapse-config/inbound-endpoints/ReintentoConsultaDBEventInboundEP.xml  
src/main/synapse-config/inbound-endpoints/ReintentoConsultaInboundEP.xml  
src/main/synapse-config/sequences/ReintentoConsultaMainErrorSeq.xml  
src/main/synapse-config/sequences/ReintentoConsultaMainSeq.xml  
src/main/synapse-config/sequences/ReintentoConsultaSunatRSAErrorSeq.xml  
src/main/synapse-config/sequences/ReintentoConsultaSunatRSASeq.xml
```

4.- ReintentoConsultaWSO2ServiceConnectorExporter: Subcomponente que permite la configuración de librerías para el consumo de mensajes de las colas en Kafka.

5.- ReintentoConsultaWSO2ServiceDockerExporter: Subcomponente que permite generar una imagen en Docker para el despliegue del componente Reintento Consulta en un entorno en nube.

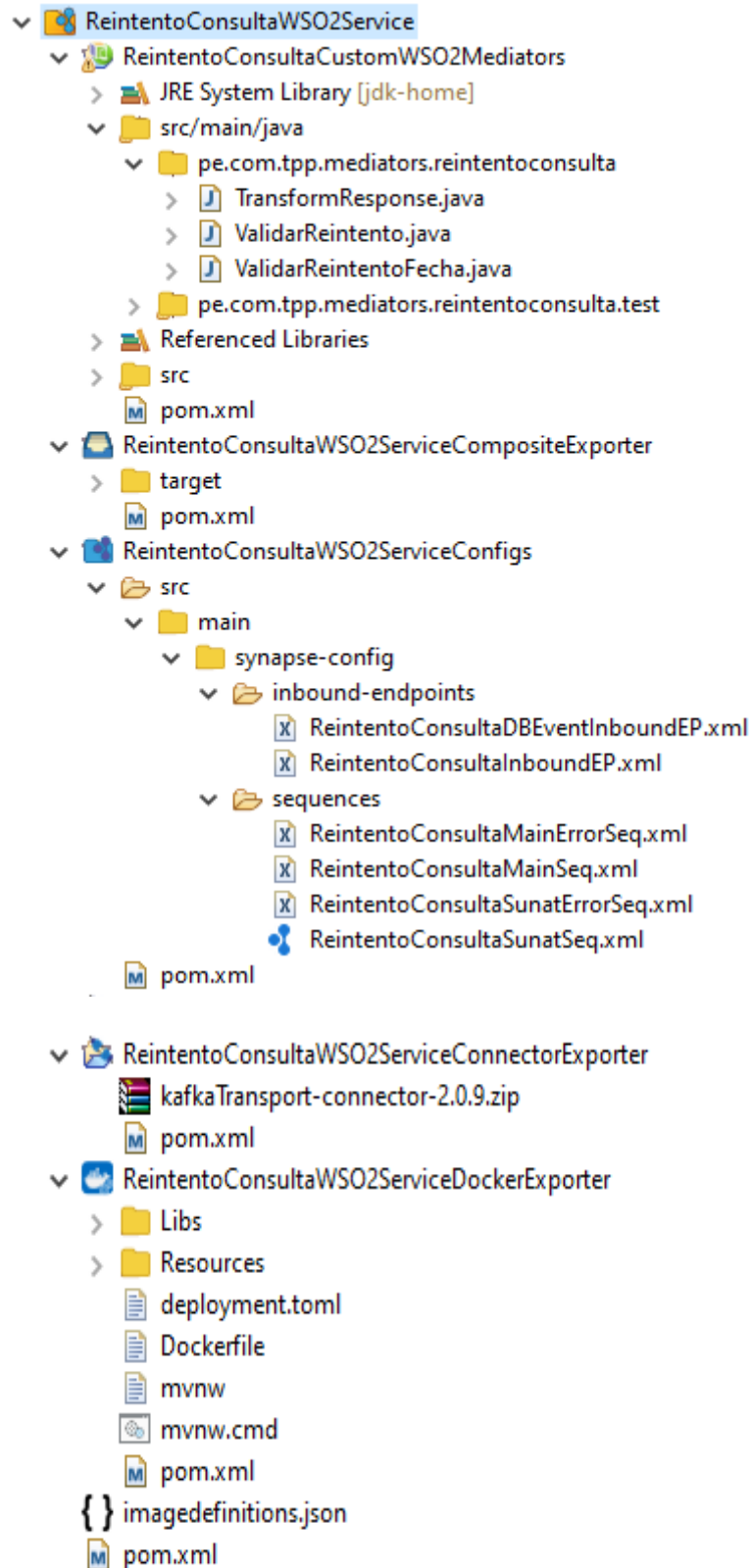


Figura 24-Estructura de proyecto del componente Reintento Consulta (Fuente: Elaboración Propia)

Desarrollar componentes de integración ESB con wso2

-Desarrollo de Componente Facade

-Entrada de evento (Inbound-Endpoints)

Se crea la entrada de evento "FacadeInboundEP.xml" que leerá los documentos de transmisión (Formato JSON) enviados a la cola de mensajes "source.tpp_transmission". Por cada lectura exitosa de documento seguirá el flujo de proceso "FacadeMainSeq" (Sequence). En caso de lectura errónea seguirá el flujo de proceso "FacadeMainErrorSeq" (Sequence).

Se define los parámetros de configuración para la entrada de evento "FacadeInboundEP.xml".

- Sequential: Lectura secuencial de los mensajes de entrada. Valor configurado: true
- Interval: Intervalo de lectura de los mensajes de entrada en milisegundos. Valor configurado: 5000
- Coordination: Ejecución en modo cluster (false) o en un solo nodo (true). Valor configurado: true
- Inbound.behavior: Comportamiento de lectura de datos. Valor configurado: pooling
- Topic.name: Nombre de la cola de mensajes de entrada para la lectura de transmisiones. Valor configurado: source.tpp_transmission
- Poll.timeout: Tiempo máximo de espera de lectura en milisegundos. Valor configurado: 1000
- Max.poll.records: Cantidad máxima de mensajes a lectura de forma periódica. Valor configurado: 5
- ContentType: Formato de mensajes de entrada. Valor configurado: application/json

Tabla 10. Código de entrada de evento FacadeInboundEP

```
<inboundEndpoint class="org.wso2.carbon.inbound.kafka.KafkaMessageConsumer" name="FacadeInboundEP"
onError="FacadeMainErrorSeq" sequence="FacadeMainSeq" suspend="false"
xmlns="http://ws.apache.org/ns/synapse">
  <parameters>
    <parameter name="sequential">true</parameter>
    <parameter name="interval">5000</parameter>
    <parameter name="coordination">true</parameter>
    <parameter name="inbound.behavior">polling</parameter>
    <parameter name="topic.name">source.tpp_transmission</parameter>
    <parameter name="poll.timeout">1000</parameter>
    <parameter name="max.poll.records">5</parameter>
    <parameter name="bootstrap.servers">${SYSTEM:KAFKA_BOOTSTRAP_SERVERS}</parameter>
    <parameter name="group.id">facadewso2service</parameter>
    <parameter name="contentType">application/json</parameter>
  </parameters>
</inboundEndpoint>
```

- Flujos de proceso (Sequence)

Se desarrollan dos flujos de procesos principales:

FacadeMainSeq: Contiene la lógica principal del flujo de proceso Facade:

- Validación de obligatoriedad de los campos del documento de transmisión (Objeto JSON)
- Transformación de campos a formato solicitado por SUNAT.
- Registro en las tablas de IRM y RSA para posterior consulta de datos.
- Notificación en casos de errores de validación.
- Registros en tablas de eventos y control del proceso.

FacadeMainErrorSeq: Contiene la lógica de control de errores del flujo de proceso "FacadeMainSeq":

- Registro de logs con el código y mensaje de error.
- Ejecuta el fin del flujo de proceso "FacadeMainSeq".

Tabla 11. Código de flujo de proceso FacadeMainSeq

```
<sequence name="FacadeMainSeq" onError="FacadeMainErrorSeq" trace="disable"
xmlns="http://ws.apache.org/ns/synapse">
  <propertyGroup description="Mensaje properties">
    <property expression="json-eval($)" name="PROP_MENSAJEJSON" scope="default" type="STRING"/>
    <property expression="json-eval($.CodTransmision)" name="PROP_CODTRANSMISION" scope="default"
type="STRING"/>
  </propertyGroup>
  <log description="Log sequence">
    <property name="LOG" value="+++++++ FACADE (MAIN SEQUENCE) ++++"/>
    <property expression="$ctx:PROP_CODTRANSMISION" name="CODIGO_TRANSMISION"/>
    <property expression="$ctx:PROP_MENSAJEJSON" name="MENSAJE_JSON"/>
  </log>
  <switch source="$ctx:PROP_CODTRANSMISION">
    <case regex="RSA">
      <log description="RSA">
        <property name="LOG" value="+++++++ RSA ++++"/>
      </log>
      <class description="Procesar Data RSA" name="pe.com.tpp.mediators.facade.ProcesarDataRSA">
        <property expression="$ctx:cpnid_salidaalmacen" name="cpnid_salidaalmacen"/>
        <property expression="$ctx:cpnid_blmastersalidaalmacen" name="cpnid_blmastersalidaalmacen"/>
        <property expression="$ctx:cpnid_tblmaster" name="cpnid_tblmaster"/>
        <property expression="$ctx:cpnid_tblcontainers" name="cpnid_tblcontainers"/>
        <property expression="$ctx:cpnid_tautorizacion" name="cpnid_tautorizacion"/>
        <property expression="$ctx:cpnid_tticket" name="cpnid_tticket"/>
        <property expression="$ctx:cpcaduanadeproceso" name="cpcaduanadeproceso"/>
        <property expression="$ctx:cpcannideclaracion" name="cpcannideclaracion"/>
        <property expression="$ctx:cpcodigoregimendeclaracion" name="cpcodigoregimendeclaracion"/>
        <property expression="$ctx:cpcnumerodeclaracion" name="cpcnumerodeclaracion"/>
        <property expression="$ctx:cpclacavehiculo" name="cpclacavehiculo"/>
        <property expression="$ctx:cpcbreveteconductor" name="cpcbreveteconductor"/>
        <property expression="$ctx:cpclasificamercancia" name="cpclasificamercancia"/>
        <property expression="$ctx:cpcidentificaciondeequipamiento"
name="cpcidentificaciondeequipamiento"/>
        <property expression="$ctx:cpcnumeroprecinto01" name="cpcnumeroprecinto01"/>
        <property expression="$ctx:cpntotalbulto" name="cpntotalbulto"/>
        <property expression="$ctx:cpnbultosaldo" name="cpnbultosaldo"/>
        <property expression="$ctx:cpnpesobruto" name="cpnpesobruto"/>
        <property expression="$ctx:cpdfechasalida" name="cpdfechasalida"/>
        <property expression="$ctx:cpcdescripcionmercancia" name="cpcdescripcionmercancia"/>
      </class>
    </case>
  </switch>
</sequence>
```

```

<property expression="$Ctx:cpnvalidapesobulto" name="cpnvalidapesobulto"/>
<property expression="$Ctx:cpnsalidaparcial" name="cpnsalidaparcial"/>
<property expression="$Ctx:cpncodigoestadoenvio" name="cpncodigoestadoenvio"/>
<property expression="$Ctx:cpcnumerocorrelativodeclaracion"
name="cpcnumerocorrelativodeclaracion"/>
<property expression="$Ctx:cpcnumautorizacion" name="cpcnumautorizacion"/>
<property expression="$Ctx:cpcnumdocuadua" name="cpcnumdocuadua"/>
<property expression="$Ctx:cpcobservacion" name="cpcobservacion"/>
<property expression="$Ctx:cpcdescripcionrespuesta" name="cpcdescripcionrespuesta"/>
<property expression="$Ctx:cpnid_envioaduana" name="cpnid_envioaduana"/>
<property expression="$Ctx:cpnid_empresasucursal" name="cpnid_empresasucursal"/>
<property expression="$Ctx:cpnid_tipoenvioconfigura" name="cpnid_tipoenvioconfigura"/>
<property expression="$Ctx:cpnid_condicioncarga" name="cpnid_condicioncarga"/>
<property expression="$Ctx:cpcrecalada" name="cpcrecalada"/>
<property expression="$Ctx:cpcnumblid" name="cpcnumblid"/>
<property expression="$Ctx:cpnnumerovolante" name="cpnnumerovolante"/>
<property expression="$Ctx:cpcnumerodam" name="cpcnumerodam"/>
<property expression="$Ctx:cpcructransportista" name="cpcructransportista"/>
<property expression="$Ctx:cpntaracamion" name="cpntaracamion"/>
<property expression="$Ctx:cpnpesoneto" name="cpnpesoneto"/>
<property expression="$Ctx:cpcmanifiesto" name="cpcmanifiesto"/>
<property expression="$Ctx:cpcnumdetalle" name="cpcnumdetalle"/>
</class>
<switch source="$Ctx:p_isdatacomplete">
<case regex="yes">
<log description="Continuar proceso">
<property name="LOG" value="+++++++ CONTINUAR PROCESO (DATOS COMPLETOS)
+++++++"/>
</log>
</case>
<default>
<log description="Detener proceso">
<property name="LOG" value="+++++++ DETENER PROCESO (DATOS INCOMPLETOS)
+++++++"/>
</log>
<dblookup description="Select config">
<connection>
<pool>
<driver>org.postgresql.Driver</driver>
<url>${SYSTEM:DB_PG_URL}</url>
<user>${SYSTEM:DB_PG_USR}</user>
<password>${SYSTEM:DB_PG_PSW}</password>
</pool>
</connection>
<statement>
<sql><![CDATA[SELECT cpccorreotalertausuario FROM tbl_sda_configuracion WHERE
cpccodtransmision = ?]]></sql>
<parameter expression="$Ctx:PROP_CODTRANSMISION" type="CHAR"/>
<result column="cpccorreotalertausuario" name="PROP_EMAIL_RECIPIENTS"/>
</statement>
</dblookup>
<log description="Email">
<property name="LOG" value="+++++++ ENVIAR EMAIL +++++++"/>
<property expression="$Ctx:PROP_EMAIL_RECIPIENTS" name="CORREO(S)"/>
</log>
<class description="Enviar Alerta" name="pe.com.tpp.mediators.facade.EnviaAlerta">
<property name="subject" value="ALERTA! Transmision RSA desestimada por contener datos nul
o vacios"/>
<property expression="$Ctx:PROP_EMAIL_CONTENT" name="content"/>
<property expression="$Ctx:PROP_EMAIL_RECIPIENTS" name="recipients"/>
<property name="user" value="$Ctx:PROP_USR"/>
<property name="password" value="$Ctx:PROP_PSW"/>
<property name="host" value="$Ctx:PROP_HOST"/>

```

```

    <property name="from" value="enviocorreo@tpp.com.pe"/>
  </class>
  <log description="End retry">
    <property name="LOG" value="+++++++ FIN DE PROCESO +++++++"/>
  </log>
  <drop description="End flow"/>
</default>
</switch>
<dbreport description="Insert transmision">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <url>${SYSTEM:DB_PG_URL}</url>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[INSERT INTO tbl_sda_rsa (cpccodtransmision, cpcmensajejson, cpnid_salidaalmacen,
cpnid_blmastersalidaalmacen, cpnid_tblmaster, cpnid_tblcontainers, cpnid_tautorizacion, cpnid_tticket,
cpcaduanadeproceto, cpcanniodeclaracion, cpccodigoregimendeclaracion, cpcnumerodeclaracion,
cpcplacavehiculo, cpcbreveteconductor, cpcclasificamercancia, cpcidentificaciondeequipamiento,
cpcnumeroprecinto01, cpntotalbulto, cpnbultosaldo, cpnpesobrito, cpdfechasalida, cpcdescripcionmercancia,
cpnvalidapesobulto, cpnsalidaparcial, cpncodigoestadoenvio, cpcnumerocorrelativodeclaracion,
cpcnumautorizacion, cpcnumdocuadua, cpcobservacion, cpcdescripcionrespuesta, cpnid_envioaduana,
cpnid_empresasucursal, cpnid_tipoenvioconfigura, cpnid_condicioncarga, cpcusuarioservicio,
cpdfechaultmodificacion, cpcrecalada, cpnumbld, cpnnumerovolante, cpcnumerodam, cpctransportista,
cpntaracamion, cpnpeoneto, cpcmanifiesto, cpcnumdetalle) VALUES (?, to_json(?::json), ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, TO_TIMESTAMP(?, 'YYYY-MM-DD"HH24:MI:SS'), ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
CURRENT_TIMESTAMP, ?, ?, ?, ?, ?, ?)]]></sql>
    <parameter expression="${ctx:PROP_CODTRANSMISION}" type="CHAR"/>
    <parameter expression="${ctx:PROP_MENSAJEJSON}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpnid_salidaalmacen}" type="INTEGER"/>
    <parameter expression="${ctx:p_cpnid_blmastersalidaalmacen}" type="INTEGER"/>
    <parameter expression="${ctx:p_cpnid_tblmaster}" type="INTEGER"/>
    <parameter expression="${ctx:p_cpnid_tblcontainers}" type="INTEGER"/>
    <parameter expression="${ctx:p_cpnid_tautorizacion}" type="INTEGER"/>
    <parameter expression="${ctx:p_cpnid_tticket}" type="INTEGER"/>
    <parameter expression="${ctx:p_cpcaduanadeproceto}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpcanniodeclaracion}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpccodigoregimendeclaracion}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpcnumerodeclaracion}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpcplacavehiculo}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpcbreveteconductor}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpcclasificamercancia}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpcidentificaciondeequipamiento}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpcnumeroprecinto01}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpntotalbulto}" type="INTEGER"/>
    <parameter expression="${ctx:p_cpnbultosaldo}" type="NUMERIC"/>
    <parameter expression="${ctx:p_cpnpesobrito}" type="NUMERIC"/>
    <parameter expression="${ctx:p_cpdfechasalida}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpcdescripcionmercancia}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpnvalidapesobulto}" type="INTEGER"/>
    <parameter expression="${ctx:p_cpnsalidaparcial}" type="INTEGER"/>
    <parameter expression="${ctx:p_cpncodigoestadoenvio}" type="INTEGER"/>
    <parameter expression="${ctx:p_cpcnumerocorrelativodeclaracion}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpcnumautorizacion}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpcnumdocuadua}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpcobservacion}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpcdescripcionrespuesta}" type="VARCHAR"/>
    <parameter expression="${ctx:p_cpnid_envioaduana}" type="INTEGER"/>
    <parameter expression="${ctx:p_cpnid_empresasucursal}" type="INTEGER"/>
    <parameter expression="${ctx:p_cpnid_tipoenvioconfigura}" type="INTEGER"/>
  </statement>
</dbreport>

```

```

    <parameter expression="$ctx:p_cpnid_condicioncarga" type="INTEGER"/>
    <parameter type="VARCHAR" value="SERV_FACADE"/>
    <parameter expression="$ctx:p_cpcrecalada" type="VARCHAR"/>
    <parameter expression="$ctx:p_cpnumblid" type="VARCHAR"/>
    <parameter expression="$ctx:p_cpnumerovolante" type="INTEGER"/>
    <parameter expression="$ctx:p_cpnumerodam" type="VARCHAR"/>
    <parameter expression="$ctx:p_cpctransportista" type="VARCHAR"/>
    <parameter expression="$ctx:p_cpntaracamion" type="NUMERIC"/>
    <parameter expression="$ctx:p_cpnpesoneto" type="NUMERIC"/>
    <parameter expression="$ctx:p_cpmanifesto" type="VARCHAR"/>
    <parameter expression="$ctx:p_cpnumdetalle" type="VARCHAR"/>
  </statement>
</dbreport>
<dblookup description="Select id transmision">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <url>${SYSTEM:DB_PG_URL}</url>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[SELECT cpnid_rsa FROM tbl_sda_rsa WHERE cpnid_tticket = ? ORDER BY cpnid_rsa
DESC LIMIT 1]]></sql>
    <parameter expression="$ctx:p_cpnid_tticket" type="INTEGER"/>
    <result column="cpnid_rsa" name="cpnid_rsa"/>
  </statement>
</dblookup>
<property description="Numero transmision" expression="$ctx:cpnid_rsa"
name="PROP_NUMTRANSMISION" scope="default" type="STRING"/>
</case>
<case regex="0121|0122">
  <log description="IRM">
    <property name="LOG" value="+++++++ IRM +++++++"/>
  </log>
  <class description="Procesar Data IRM" name="pe.com.tpp.mediators.facade.ProcesarDataIRM">
    <property expression="$ctx:cpnidviajesfecha" name="cpnidviajesfecha"/>
    <property expression="$ctx:cpdfecleg" name="cpdfecleg"/>
    <property expression="$ctx:cpcnumeroviaje" name="cpcnumeroviaje"/>
    <property expression="$ctx:cpcmanifesto" name="cpcmanifesto"/>
    <property expression="$ctx:cpncantidadbultos" name="cpncantidadbultos"/>
    <property expression="$ctx:cpnpesobruto" name="cpnpesobruto"/>
    <property expression="$ctx:cpncantidadbultosmalos" name="cpncantidadbultosmalos"/>
    <property expression="$ctx:cpnpesomalos" name="cpnpesomalos"/>
    <property expression="$ctx:cpdfechaoperacion" name="cpdfechaoperacion"/>
    <property expression="$ctx:cpnid_conocimientoid" name="cpnid_conocimientoid"/>
    <property expression="$ctx:cpcnrodetalle" name="cpcnrodetalle"/>
    <property expression="$ctx:cpcnrodocumentotransporte" name="cpcnrodocumentotransporte"/>
    <property expression="$ctx:cpnpasarlcl" name="cpnpasarlcl"/>
    <property expression="$ctx:cpctipo" name="cpctipo"/>
    <property expression="$ctx:cpccierrebl" name="cpccierrebl"/>
    <property expression="$ctx:cpnid_contenedorid" name="cpnid_contenedorid"/>
    <property expression="$ctx:cpcnumerocontenedor" name="cpcnumerocontenedor"/>
    <property expression="$ctx:cpnindicasini" name="cpnindicasini"/>
    <property expression="$ctx:cpnid_envioaduana" name="cpnid_envioaduana"/>
    <property expression="$ctx:cpcestadoenvio" name="cpcestadoenvio"/>
    <property expression="$ctx:cpcinformacionmensaje" name="cpcinformacionmensaje"/>
    <property expression="$ctx:cpncantidadbultosmanifestado" name="cpncantidadbultosmanifestado"/>
    <property expression="$ctx:cpccodigoaduana" name="cpccodigoaduana"/>
    <property expression="$ctx:cpnid_tipoenvioconfigura" name="cpnid_tipoenvioconfigura"/>
    <property expression="$ctx:cpnid_condicioncarga" name="cpnid_condicioncarga"/>
    <property expression="$ctx:cpnid_empresasucursal" name="cpnid_empresasucursal"/>
  </class>

```

```

    <property expression="$ctx:cpcevisor" name="cpcevisor"/>
    <property expression="$ctx:cpctransvisor" name="cpctransvisor"/>
  </class>
  <switch source="$ctx:p_isdatacomplete">
    <case regex="yes">
      <log description="Continuar proceso">
        <property name="LOG" value="+++++++ CONTINUAR PROCESO (DATOS COMPLETOS)
++++++"/>
      </log>
    </case>
    <default>
      <log description="Detener proceso">
        <property name="LOG" value="+++++++ DETENER PROCESO (DATOS INCOMPLETOS)
++++++"/>
      </log>
      <dblookup description="Select config">
        <connection>
          <pool>
            <driver>org.postgresql.Driver</driver>
            <url>${SYSTEM:DB_PG_URL}</url>
            <user>${SYSTEM:DB_PG_USR}</user>
            <password>${SYSTEM:DB_PG_PSW}</password>
          </pool>
        </connection>
        <statement>
          <sql><![CDATA[SELECT cpccorreotalertausuario FROM tbl_sda_configuracion WHERE
cpccodtransmision = ?]]></sql>
          <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
          <result column="cpccorreotalertausuario" name="PROP_EMAIL_RECIPIENTS"/>
        </statement>
      </dblookup>
      <log description="Email">
        <property name="LOG" value="+++++++ ENVIAR EMAIL +++++++"/>
        <property expression="$ctx:PROP_EMAIL_RECIPIENTS" name="CORREO(S)"/>
      </log>
      <class description="Enviar Alerta" name="pe.com.tpp.mediators.facade.EnviaAlerta">
        <property name="subject" value="ALERTA! Transmision IRM desestimada por contener datos nulos
o vacios"/>
        <property expression="$ctx:PROP_EMAIL_CONTENT" name="content"/>
        <property expression="$ctx:PROP_EMAIL_RECIPIENTS" name="recipients"/>
        <property name="user" value="$ctx:PROP_USR"/>
        <property name="password" value="$ctx:PROP_PSW"/>
        <property name="host" value="$ctx:PROP_HOST"/>
        <property name="from" value="enviocorreo@tpp.com.pe"/>
      </class>
      <log description="End retry">
        <property name="LOG" value="+++++++ FIN DE PROCESO +++++++"/>
      </log>
      <drop description="End flow"/>
    </default>
  </switch>
  <dbreport description="Insert transmision">
    <connection>
      <pool>
        <driver>org.postgresql.Driver</driver>
        <url>${SYSTEM:DB_PG_URL}</url>
        <user>${SYSTEM:DB_PG_USR}</user>
        <password>${SYSTEM:DB_PG_PSW}</password>
      </pool>
    </connection>
    <statement>
      <sql><![CDATA[INSERT INTO tbl_sda_irm (cpccodtransmision, cpcmensajejson, cpnidtviajesfecha,
cpdfechallegada, cpnumeroviaje, cpcrecalada, cpcanhomanifiesto, cpnumeromanifiesto, cpncantidadbultos,

```

```

cpnpesobroto, cpncantidadbultosmalos, cpnpesomalos, cpdfechaoperacion, cpnid_conocimiento, cpcnrodetalle,
cpcnrodokumentotransportemaster, cpcnrodokumentotransporte, cpnpasarlcl, cpctipo, cpccierrebl,
cpnid_contenedor, cpnumerocontenedor, cpnindicasini, cpnindicasinicontacto, cpcnroactasini,
cpnid_envioaduana, cpcestadoenvio, cpcinformacionmensaje, cpncantidadbultosmanifestado, cpccodigoaduana,
cpnid_tipoenvioconfigura, cpnid_condicioncarga, cpnid_empresasucursal, cpemisor, cpctransmisor,
cpcusarioservicio, cpdfechaultmodificacion) VALUES (?, to_json(?:json), ?, TO_TIMESTAMP(?, 'YYYY-MM-
DD"T"HH24:MI:SS'), ?, ?, ?, ?, ?, ?, TO_TIMESTAMP(?, 'YYYY-MM-DD"T"HH24:MI:SS'), ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, CURRENT_TIMESTAMP))]></sql>
  <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
  <parameter expression="$ctx:PROP_MENSAJEJSON" type="VARCHAR"/>
  <parameter expression="$ctx:p_cpnidviajesfecha" type="INTEGER"/>
  <parameter expression="$ctx:p_cpdfecleg" type="VARCHAR"/>
  <parameter expression="$ctx:p_cpcnumeroviaje" type="VARCHAR"/>
  <parameter expression="$ctx:p_cpcrecalada" type="VARCHAR"/>
  <parameter expression="$ctx:p_cpcanhomanifiesto" type="CHAR"/>
  <parameter expression="$ctx:p_cpnummeromanifiesto" type="INTEGER"/>
  <parameter expression="$ctx:p_cpncantidadbultos" type="INTEGER"/>
  <parameter expression="$ctx:p_cpnpesobroto" type="NUMERIC"/>
  <parameter expression="$ctx:p_cpncantidadbultosmalos" type="INTEGER"/>
  <parameter expression="$ctx:p_cpnpesomalos" type="NUMERIC"/>
  <parameter expression="$ctx:p_cpdfechaoperacion" type="VARCHAR"/>
  <parameter expression="$ctx:p_cpnid_conocimientoid" type="INTEGER"/>
  <parameter expression="$ctx:p_cpcnrodetalle" type="VARCHAR"/>
  <parameter expression="$ctx:p_cpcnrodokumentotransporte" type="VARCHAR"/>
  <parameter expression="$ctx:p_cpnpasarlcl" type="INTEGER"/>
  <parameter expression="$ctx:p_cpctipo" type="VARCHAR"/>
  <parameter expression="$ctx:p_cpccierrebl" type="VARCHAR"/>
  <parameter expression="$ctx:p_cpnid_contenedorid" type="INTEGER"/>
  <parameter expression="$ctx:p_cpcnumerocontenedor" type="VARCHAR"/>
  <parameter expression="$ctx:p_cpnindicasini" type="INTEGER"/>
  <parameter expression="$ctx:p_cpnid_envioaduana" type="INTEGER"/>
  <parameter expression="$ctx:p_cpcestadoenvio" type="VARCHAR"/>
  <parameter expression="$ctx:p_cpcinformacionmensaje" type="VARCHAR"/>
  <parameter expression="$ctx:p_cpncantidadbultosmanifestado" type="INTEGER"/>
  <parameter expression="$ctx:p_cpccodigoaduana" type="CHAR"/>
  <parameter expression="$ctx:p_cpnid_tipoenvioconfigura" type="INTEGER"/>
  <parameter expression="$ctx:p_cpnid_condicioncarga" type="INTEGER"/>
  <parameter expression="$ctx:p_cpnid_empresasucursal" type="INTEGER"/>
  <parameter type="VARCHAR" value="SERV_FACADE"/>
</statement>
</dbreport>
<dblookup description="Select id transmision">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <url>${SYSTEM:DB_PG_URL}</url>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[SELECT cpnid_irm FROM tbl_sda_irm WHERE cpnidviajesfecha = ? ORDER BY
cpnid_irm DESC LIMIT 1]]></sql>
    <parameter expression="$ctx:p_cpnidviajesfecha" type="INTEGER"/>
    <result column="cpnid_irm" name="cpnid_transmisionirm"/>
  </statement>
</dblookup>
  <property description="Numero transmision" expression="$ctx:cpnid_transmisionirm"
name="PROP_NUMTRANSMISION" scope="default" type="STRING"/>
</case>
<default>
  <drop description="Drop message"/>
</default>

```



```

</switch>
<dbreport description="Insert control">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <url>${SYSTEM:DB_PG_URL}</url>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[INSERT INTO tbl_sda_control (cpnnumerotransmission, cpccodtransmission, cpnid_evento,
cpcusuario_servicio) VALUES (?, ?, ?, ?)]]></sql>
    <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
    <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
    <parameter type="INTEGER" value="1"/>
    <parameter type="VARCHAR" value="SERV_FACADE"/>
  </statement>
</dbreport>
<log description="Log envio">
  <property name="LOG" value="+++++++ ENVIAR A COLA DE TRANSMISION ++++"/>
</log>
<payloadFactory description="Payload" media-type="json">
  <format>
    {"numero_transmission": "$1", "codigo_transmission": "$2"}</format>
  <args>
    <arg evaluator="xml" expression="$ctx:PROP_NUMTRANSMISION"/>
    <arg evaluator="xml" expression="$ctx:PROP_CODTRANSMISION"/>
  </args>
</payloadFactory>
<log description="Log json payload" level="full"/>
<property description="Content type" name="ContentType" scope="axis2" type="STRING"
value="application/json"/>
<script description="Environment variables"
language="js"><![CDATA[mc.setProperty("PROP_KAFKA_BOOTSTRAP_SERVERS",
java.lang.System.getenv("KAFKA_BOOTSTRAP_SERVERS"));]]></script>
<kafkaTransport.init>
  <bootstrapServers>{$ctx:PROP_KAFKA_BOOTSTRAP_SERVERS}</bootstrapServers>
</kafkaTransport.init>
<kafkaTransport.publishMessages>
  <topic>tpp_transmission_envio</topic>
</kafkaTransport.publishMessages>
<drop description="End flow"/>
</sequence>

```

Tabla 12. Código de flujo de proceso FacadeMainErrorSeq

```

<sequence name="FacadeMainErrorSeq" trace="disable" xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ FACADE (MAIN ERROR SEQUENCE) ++++"/>
    <property expression="get-property('ERROR_CODE')" name="ERROR_CODE"/>
    <property expression="get-property('ERROR_MESSAGE')" name="ERROR_MESSAGE"/>
    <property expression="get-property('ERROR_DETAIL')" name="ERROR_DETAIL"/>
    <property expression="get-property('ERROR_EXCEPTION')" name="ERROR_EXCEPTION"/>
  </log>
  <drop description="End flow"/>
</sequence>

```

-Desarrollo de Componente Transmisión

-Entrada de evento (Inbound-Endpoints)

Se crea la entrada de evento "TransmisionInboundEP.xml" que leerá las entradas (Formato JSON) enviados a la cola de mensajes "tpp_transmision_envio". Por cada lectura exitosa de documento seguirá el flujo de proceso "TransmisionMainSeq" (Sequence). En caso de lectura errónea seguirá el flujo de proceso "TransmisionMainErrorSeq" (Sequence).

Se define los parámetros de configuración para la entrada de evento "TransmisionInboundEP.xml".

- Sequential: Lectura secuencial de los mensajes de entrada. Valor configurado: true
- Interval: Intervalo de lectura de los mensajes de entrada en milisegundos. Valor configurado: 5000
- Coordination: Ejecución en modo cluster (false) o en un solo nodo (true). Valor configurado: true
- Inbound.behavior: Comportamiento de lectura de datos. Valor configurado: pooling
- Topic.name: Nombre de la cola de mensajes de entrada para la lectura de transmisiones. Valor configurado: tpp_transmision_envio
- Poll.timeout: Tiempo máximo de espera de lectura en milisegundos. Valor configurado: 1000
- Max.poll.records: Cantidad máxima de mensajes a lectura de forma periódica. Valor configurado: 5
- ContentType: Formato de mensajes de entrada. Valor configurado: application/json

Tabla 13. Código de entrada de evento TransmisionInboundEP

```
<inboundEndpoint class="org.wso2.carbon.inbound.kafka.KafkaMessageConsumer"
name="TransmisionInboundEP" onError="TransmisionMainErrorSeq" sequence="TransmisionMainSeq"
suspend="false" xmlns="http://ws.apache.org/ns/synapse">
  <parameters>
    <parameter name="sequential">true</parameter>
    <parameter name="interval">5000</parameter>
    <parameter name="coordination">true</parameter>
    <parameter name="inbound.behavior">polling</parameter>
    <parameter name="topic.name">tpp_transmision_envio</parameter>
    <parameter name="poll.timeout">1000</parameter>
    <parameter name="max.poll.records">5</parameter>
    <parameter name="bootstrap.servers">${SYSTEM:KAFKA_BOOTSTRAP_SERVERS}</parameter>
    <parameter name="group.id">transmisionwso2service</parameter>
    <parameter name="contentType">application/json</parameter>
  </parameters>
</inboundEndpoint>
```

- Flujos de proceso (Sequence)

Se desarrollan dos flujos de procesos principales y 4 flujos secundarios:

TransmisionMainSeq: Contiene la lógica principal del flujo de proceso Transmisión:

- Validación del límite de tiempo de envío según configuración.
- Registros en tablas de eventos y control del proceso.
- Switch de decisión: Si es tipo IRM delegar la continuidad al subproceso "TransmisionSunatIRMSeq.xml". Si es tipo RSA delegar la continuidad al subproceso "TransmisionSunatRSASeq.xml".

TransmisionMainErrorSeq: Contiene la lógica de control de errores del flujo de proceso "TransmisionMainSeq":

- Registro de logs con el código y mensaje de error.
- Ejecuta el fin del flujo de proceso "TransmisionMainSeq".

TransmisionSunatIRMSeq: Contiene la lógica principal del flujo de proceso para IRM:

- Lectura de plantilla IRM para armado de documento de transmisión.
- Creación de cliente de comunicación para enviar el documento de transmisión a Web Services de recepción de documentos IRM de SUNAT.
- Lectura de respuesta (Fecha de recepción, Nro de Ticket) de Web Services de SUNAT.
- Registros en tablas de eventos y control del proceso.

TransmisionSunatIRMErrorSeq: Contiene la lógica de control de errores del flujo de proceso "TransmisionSunatIRMSeq":

- Registro de logs con el código y mensaje de error.
- Ejecuta el fin del flujo de proceso "TransmisionMainSeq".

TransmisionSunatRSASeq: Contiene la lógica principal del flujo de proceso para RSA:

- Lectura de plantilla RSA para armado de documento de transmisión.
- Creación de cliente de comunicación para enviar el documento de transmisión a Web Services de recepción de documentos RSA de SUNAT.
- Lectura de respuesta (Estado de transmisión) de Web Services de SUNAT.
- Registros en tablas de eventos y control del proceso.

TransmisionSunatRSAErrorSeq: Contiene la lógica de control de errores del flujo de proceso "TransmisionSunatRSASeq":

- Registro de logs con el código y mensaje de error.
- Ejecuta el fin del flujo de proceso "TransmisionMainSeq".

Tabla 14. Código de flujo de proceso TransmisionMainSeq

```
<sequence name="TransmisionMainSeq" onError="TransmisionMainErrorSeq" trace="disable"
xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ TRANSMISION (MAIN SEQUENCE) +++++++"/>
    <property expression="json-eval($)" name="MENSAJE_JSON"/>
  </log>
</sequence>
```

```

</log>
<propertyGroup description="Transmission properties">
  <property expression="//numero_transmision" name="PROP_NUMTRANSMISION" scope="default"
type="STRING"/>
  <property expression="//codigo_transmision" name="PROP_CODTRANSMISION" scope="default"
type="STRING"/>
</propertyGroup>
<dbreport description="Insert control">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <url>${SYSTEM:DB_PG_URL}</url>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[INSERT INTO tbl_sda_control (cpnnumerotransmision, cpccodtransmision, cpnid_evento,
cpcusuario_servicio) VALUES (?, ?, ?, ?)]]></sql>
    <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
    <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
    <parameter type="INTEGER" value="2"/>
    <parameter type="VARCHAR" value="SERV_TRANSMISION"/>
  </statement>
</dbreport>
<dblookup description="Select config">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <url>${SYSTEM:DB_PG_URL}</url>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[SELECT cpnlimiteplazoenvio, cpcusuariows, cpcclavews, cpccodtransaccionoma FROM
tbl_sda_configuracion WHERE cpccodtransmision = ?]]></sql>
    <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
    <result column="cpcclavews" name="cpcclavews"/>
    <result column="cpccodtransaccionoma" name="cpccodtransaccionoma"/>
    <result column="cpcusuariows" name="cpcusuariows"/>
    <result column="cpnlimiteplazoenvio" name="cpnlimiteplazoenvio"/>
  </statement>
</dblookup>
<propertyGroup description="Properties config">
  <property expression="get-property('cpnlimiteplazoenvio')" name="PROP_LIMITEENVIO" scope="default"
type="STRING"/>
  <property expression="get-property('cpcusuariows')" name="PROP_USUARIOWS" scope="default"
type="STRING"/>
  <property expression="get-property('cpcclavews')" name="PROP_CLAVEWS" scope="default"
type="STRING"/>
  <property expression="get-property('cpccodtransaccionoma')" name="PROP_CODOMA" scope="default"
type="STRING"/>
</propertyGroup>
<switch source="$ctx:PROP_CODTRANSMISION">
  <case regex="0121|0122">
    <log description="IRM">
      <property name="LOG" value="+++++++ IRM +++++++"/>
    </log>
    <dblookup description="Select data transmision">
      <connection>
        <pool>
          <driver>org.postgresql.Driver</driver>

```

```

        <url>${SYSTEM:DB_PG_URL}</url>
        <user>${SYSTEM:DB_PG_USR}</user>
        <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
</connection>
<statement>
    <sql><![CDATA[SELECT cpccodtransmision, TO_CHAR(cpdfechaoperacion, 'YYYYMMDDHH24MISS') AS
cpdfechaoperacion, cpcanhomanifiesto, cpnnumeromanifiesto, cpcnrodetalle, cpncantidadbultos, cpnpesobruto,
cpcnrodokumentotransporte, cpncantidadbultosmalos FROM tbl_sda_irm WHERE cpnid_irm = ?]]></sql>
    <parameter expression="Ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
    <result column="cpncantidadbultosmalos" name="cpncantidadbultosmalos"/>
    <result column="cpccodtransmision" name="cpccodtransmision"/>
    <result column="cpncantidadbultos" name="cpncantidadbultos"/>
    <result column="cpnpesobruto" name="cpnpesobruto"/>
    <result column="cpcnrodokumentotransporte" name="cpcnrodokumentotransporte"/>
    <result column="cpcanhomanifiesto" name="cpcanhomanifiesto"/>
    <result column="cpcnrodetalle" name="cpcnrodetalle"/>
    <result column="cpdfechaoperacion" name="cpdfechaoperacion"/>
    <result column="cpnnumeromanifiesto" name="cpnnumeromanifiesto"/>
</statement>
</dblookup>
<dblookup description="Select correlativo">
    <connection>
        <pool>
            <driver>org.postgresql.Driver</driver>
            <url>${SYSTEM:DB_PG_URL}</url>
            <user>${SYSTEM:DB_PG_USR}</user>
            <password>${SYSTEM:DB_PG_PSW}</password>
        </pool>
    </connection>
    <statement>
        <sql><![CDATA[SELECT CONCAT(cpnanhocorrelativo, cpnseriecorrelativo) AS correlativo FROM
tbl_sda_correlativo WHERE cpnnumerotransmision = ? AND cpccodtransmision = ?]]></sql>
        <parameter expression="Ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
        <parameter expression="Ctx:PROP_CODTRANSMISION" type="CHAR"/>
        <result column="correlativo" name="correlativo"/>
    </statement>
</dblookup>
<class description="Build IRM XML" name="pe.com.tpp.transmisiones.BuildIRMXML">
    <property expression="get-property('LE_TRANSMISION_IRM')" name="originalIRMXML"/>
    <property expression="get-property('cpccodtransmision')" name="cpccodtransmision"/>
    <property expression="get-property('cpdfechaoperacion')" name="cpdfechaoperacion"/>
    <property expression="get-property('cpcanhomanifiesto')" name="cpcanhomanifiesto"/>
    <property expression="get-property('cpnnumeromanifiesto')" name="cpnnumeromanifiesto"/>
    <property expression="get-property('cpcnrodetalle')" name="cpcnrodetalle"/>
    <property expression="get-property('cpncantidadbultos')" name="cpncantidadbultos"/>
    <property expression="get-property('cpncantidadbultosmalos')" name="cpncantidadbultosmalos"/>
    <property expression="get-property('correlativo')" name="correlativo"/>
</class>
<sequence key="TransmissionSunatIRMSeq"/>
<drop description="End flow"/>
</case>
<case regex="RSA">
    <log description="RSA">
        <property name="LOG" value="+++++++ RSA +++++++"/>
    </log>
    <dblookup description="Minutes difference">
        <connection>
            <pool>
                <driver>org.postgresql.Driver</driver>
                <url>${SYSTEM:DB_PG_URL}</url>
                <user>${SYSTEM:DB_PG_USR}</user>
                <password>${SYSTEM:DB_PG_PSW}</password>
            </pool>
        </connection>
    </dblookup>

```

```

    </pool>
  </connection>
  <statement>
    <sql><![CDATA[SELECT (DATE_PART('day', timezone('America/Lima', CURRENT_TIMESTAMP) -
cpdfechasalida) * 24 + DATE_PART('hour', timezone('America/Lima', CURRENT_TIMESTAMP) - cpdfechasalida)) *
60 + DATE_PART('minute', timezone('America/Lima', CURRENT_TIMESTAMP) - cpdfechasalida) AS diff_min,
(DATE_PART('day', timezone('America/Lima', CURRENT_TIMESTAMP) - cpdfechasalida) * 24 + DATE_PART('hour',
timezone('America/Lima', CURRENT_TIMESTAMP) - cpdfechasalida)) * 60 + DATE_PART('minute',
timezone('America/Lima', CURRENT_TIMESTAMP) - cpdfechasalida) < ? AS result_diff_min FROM tbl_sda_rsa
WHERE cpnid_rsa = ?]]></sql>
    <parameter expression="$ctx:PROP_LIMITEENVIO" type="INTEGER"/>
    <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
    <result column="diff_min" name="diff_min"/>
    <result column="result_diff_min" name="result_diff_min"/>
  </statement>
</dblookup>
<property description="Minutes" expression="get-property('diff_min')" name="PROP_DIFFMIN"
scope="default" type="STRING"/>
<property description="Result minutes" expression="get-property('result_diff_min')"
name="PROP_RESULTDIFFMIN" scope="default" type="STRING"/>
<log description="Log minutes">
  <property name="LOG" value="+++++++ MINUTOS PASADOS ++++"/>
  <property expression="$ctx:PROP_DIFFMIN" name="MINUTOS"/>
  <property expression="$ctx:PROP_RESULTDIFFMIN" name="CONTINUAR"/>
</log>
<switch source="$ctx:PROP_RESULTDIFFMIN">
  <case regex="true">
    <log description="Continuar">
      <property name="LOG" value="+++++++ CONTINUAR CON ENVIO ++++"/>
    </log>
    <dblookup description="Select data transmision">
      <connection>
        <pool>
          <driver>org.postgresql.Driver</driver>
          <url>${SYSTEM:DB_PG_URL}</url>
          <user>${SYSTEM:DB_PG_USR}</user>
          <password>${SYSTEM:DB_PG_PSW}</password>
        </pool>
      </connection>
      <statement>
        <sql><![CDATA[SELECT cpccodtransmision, COALESCE(cpcaduanadeprocso,") as
cpcaduanadeprocso, COALESCE(cpcannideclaracion,") as cpcannideclaracion,
COALESCE(cpccodigoregimendeclaracion,") as cpccodigoregimendeclaracion,
COALESCE(cpcnumerodeclaracion,") as cpcnumerodeclaracion, cpcbreveteconductor, cpclacavehiculo,
cpclasificamercancia, cpcentificaciondelequipamiento, COALESCE(cpcnumeroprecinto01,") as
cpcnumeroprecinto01, cpnpesobruuto, TO_CHAR(cpdfechasalida, 'MM/DD/YYYY HH24:MI:SS') AS cpdfechasalida,
cpnsalidaparcial, cpnvalidapesobulto, COALESCE(cpcdescripcionmercancia,") as cpcdescripcionmercancia,
cpntotalbulto FROM tbl_sda_rsa WHERE cpnid_rsa = ?]]></sql>
        <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
        <result column="cpcaduanadeprocso" name="cpcaduanadeprocso"/>
        <result column="cpnvalidapesobulto" name="cpnvalidapesobulto"/>
        <result column="cpccodigoregimendeclaracion" name="cpccodigoregimendeclaracion"/>
        <result column="cpnpesobruuto" name="cpnpesobruuto"/>
        <result column="cpclasificamercancia" name="cpclasificamercancia"/>
        <result column="cpdfechasalida" name="cpdfechasalida"/>
        <result column="cpclacavehiculo" name="cpclacavehiculo"/>
        <result column="cpcannideclaracion" name="cpcannideclaracion"/>
        <result column="cpccodtransmision" name="cpccodtransmision"/>
        <result column="cpnsalidaparcial" name="cpnsalidaparcial"/>
        <result column="cpcnumerodeclaracion" name="cpcnumerodeclaracion"/>
        <result column="cpcdescripcionmercancia" name="cpcdescripcionmercancia"/>
        <result column="cpntotalbulto" name="cpntotalbulto"/>
        <result column="cpcbreveteconductor" name="cpcbreveteconductor"/>
      </statement>
    </dblookup>
  </case>
</switch>

```

```

        <result column="cpcidentificaciondeequipamiento" name="cpcidentificaciondeequipamiento"/>
        <result column="cpcnumeroprecinto01" name="cpcnumeroprecinto01"/>
    </statement>
</dblookup>
<class description="Build RSA XML" name="pe.com.tpp.transmisiones.BuildRSAXML">
    <property expression="get-property('LE_TRANSMISION_RSA')" name="originalRSAXML"/>
    <property expression="get-property('cpccodtransmision')" name="cpccodtransmision"/>
    <property expression="get-property('cpcaduanadeproceto')" name="cpcaduanadeproceto"/>
    <property expression="get-property('cpcanniodeclaracion')" name="cpcanniodeclaracion"/>
    <property expression="get-property('cpccodigoregimendeclaracion')"
name="cpccodigoregimendeclaracion"/>
    <property expression="get-property('cpcnumerodeclaracion')" name="cpcnumerodeclaracion"/>
    <property expression="get-property('cpcbreveteconductor')" name="cpcbreveteconductor"/>
    <property expression="get-property('cpcplacavehiculo')" name="cpcplacavehiculo"/>
    <property expression="get-property('cpcclasificamercancia')" name="cpcclasificamercancia"/>
    <property expression="get-property('cpcidentificaciondeequipamiento')"
name="cpcidentificaciondeequipamiento"/>
    <property expression="get-property('cpcnumeroprecinto01')" name="cpcnumeroprecinto01"/>
    <property expression="get-property('cpnpesobruto')" name="cpnpesobruto"/>
    <property expression="get-property('cpdfechasalida')" name="cpdfechasalida"/>
    <property expression="get-property('cpnsalidaparcial')" name="cpnsalidaparcial"/>
    <property expression="get-property('cpnvalidapesobulto')" name="cpnvalidapesobulto"/>
    <property expression="get-property('cpcdescripcionmercancia')"
name="cpcdescripcionmercancia"/>
    <property expression="get-property('cpntotalbulto')" name="cpntotalbulto"/>
</class>
</case>
<default>
    <log description="Desestimar">
        <property name="LOG" value="+++++++ DETENER ENVIO +++++++"/>
    </log>
    <dbreport description="Insert control">
        <connection>
            <pool>
                <driver>org.postgresql.Driver</driver>
                <url>${SYSTEM:DB_PG_URL}</url>
                <user>${SYSTEM:DB_PG_USR}</user>
                <password>${SYSTEM:DB_PG_PSW}</password>
            </pool>
        </connection>
        <statement>
            <sql><![CDATA[INSERT INTO tbl_sda_control (cpnumero transmision, cpccod transmision,
cpnid_evento, cpcusuario_servicio) VALUES (?, ?, ?, ?)]]></sql>
            <parameter expression="${ctx:PROP_NUMTRANSMISION}" type="INTEGER"/>
            <parameter expression="${ctx:PROP_CODTRANSMISION}" type="CHAR"/>
            <parameter type="INTEGER" value="6"/>
            <parameter type="VARCHAR" value="SERV_TRANSMISION"/>
        </statement>
    </dbreport>
    <drop description="End flow"/>
</default>
</switch>
<sequence key="TransmisionSunatRSASeq"/>
<drop description="End flow"/>
</case>
<default>
    <log description="Default">
        <property name="LOG" value="+++++++"/>
    </log>
    <drop description="End flow"/>
</default>
</switch>
</sequence>

```

Tabla 15. Código de flujo de proceso TransmisionMainErrorSeq

```
<sequence name="TransmisionMainErrorSeq" trace="disable" xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ TRANSMISION (MAIN ERROR SEQUENCE) +++++++"/>
    <property expression="get-property('ERROR_CODE')" name="ERROR_CODE"/>
    <property expression="get-property('ERROR_MESSAGE')" name="ERROR_MESSAGE"/>
    <property expression="get-property('ERROR_DETAIL')" name="ERROR_DETAIL"/>
    <property expression="get-property('ERROR_EXCEPTION')" name="ERROR_EXCEPTION"/>
  </log>
  <drop description="End flow"/>
</sequence>
```

Tabla 16. Código de flujo de proceso TransmisionSunatIRMSeq

```
<sequence name="TransmisionSunatIRMSeq" onError="TransmisionSunatIRMErrorSeq" trace="disable"
xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ TRANSMISION SUNAT OMA (SEQUENCE) +++++++"/>
    <property expression="$ctx:PROP_NUMTRANSMISION" name="NUMERO_TRANSMISION"/>
    <property expression="$ctx:PROP_CODTRANSMISION" name="CODIGO_TRANSMISION"/>
  </log>
  <payloadFactory description="Create payload" media-type="xml">
    <format>
      <soapenv:Envelope xmlns:ser="http://services.sigad.sunat.gob.pe"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
        <soapenv:Body>
          <recibirArchivo xmlns="http://services.sigad.sunat.gob.pe">
            <numeroTransaccion xmlns="">$3</numeroTransaccion>
            <informacionArchivo xmlns="">$4</informacionArchivo>
          </recibirArchivo>
        </soapenv:Body>
      </soapenv:Envelope>
    </format>
    <args>
      <arg evaluator="xml" expression="$ctx:PROP_CODOMA"/>
      <arg evaluator="xml" expression="$ctx:base64String"/>
    </args>
  </payloadFactory>
  <property description="Petición envío" expression="/*" name="PROP_PETICIONENVIO" scope="default"
type="STRING"/>
  <dbreport description="Update monitoreo">
    <connection>
      <pool>
        <driver>org.postgresql.Driver</driver>
        <url>${SYSTEM:DB_PG_URL}</url>
        <user>${SYSTEM:DB_PG_USR}</user>
        <password>${SYSTEM:DB_PG_PSW}</password>
      </pool>
    </connection>
    <statement>
      <sql><![CDATA[UPDATE tbl_sda_monitoreo SET cpcarchivoenvioxml = ?, cpcadenabase64 = ?,
cpcpeticionenvioxml = ? WHERE cpnumerotransmision = ? AND cpcodtransmision = ?]]></sql>
      <parameter expression="$ctx:inlineXML" type="VARCHAR"/>
      <parameter expression="$ctx:base64String" type="VARCHAR"/>
      <parameter expression="$ctx:PROP_PETICIONENVIO" type="VARCHAR"/>
      <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
      <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
    </statement>
  </dbreport>
</sequence>
```



```

    </statement>
  </dbreport>
  <property description="Content type" name="ContentType" scope="axis2" type="STRING" value="text/xml"/>
  <call>
    <endpoint>
      <wsdl port="ReceptorWebServiceServiceImplPort" service="ReceptorService.htm"
uri="$SYSTEM:WSDL_URI">
        </wsdl>
      </endpoint>
    </call>
    <log description="Log response payload" level="full"/>
    <dbreport description="Update monitoreo">
      <connection>
        <pool>
          <driver>org.postgresql.Driver</driver>
          <url>$SYSTEM:DB_PG_URL</url>
          <user>$SYSTEM:DB_PG_USR</user>
          <password>$SYSTEM:DB_PG_PSW</password>
        </pool>
      </connection>
      <statement>
        <sql><![CDATA[UPDATE tbl_sda_monitoreo SET cpcacuseenvioxml = ?, cpcanhoenvio = ?,
cpdfecharecepcionenvio = TO_TIMESTAMP( ?, 'YYYY-MM-DD"T"HH24:MI:SS'), cpcticketenvio = ? WHERE
cpnnumerotransmision = ? AND cpccodtransmision = ?]]></sql>
        <parameter expression="/" type="VARCHAR"/>
        <parameter expression="//*[local-name()='anhoEnvio']/text()" type="CHAR"/>
        <parameter expression="//*[local-name()='fechaRecepcion']/text()" type="VARCHAR"/>
        <parameter expression="//*[local-name()='ticketEnvio']/text()" type="VARCHAR"/>
        <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
        <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
      </statement>
    </dbreport>
    <property description="Ticket envio" expression="//*[local-name()='ticketEnvio']/text()"
name="PROP_TICKETENVIO" scope="default" type="STRING"/>
    <property description="Anho envio" expression="//*[local-name()='anhoEnvio']/text()"
name="PROP_ANHOENVIO" scope="default" type="STRING"/>
    <payloadFactory description="Create kafka payload" media-type="json">
      <format>
        {"numero_transmision": "$1", "codigo_transmision": "$2", "ticket_envio": "$3",
"anho_envio": "$4"}</format>
      <args>
        <arg evaluator="xml" expression="$ctx:PROP_NUMTRANSMISION"/>
        <arg evaluator="xml" expression="$ctx:PROP_CODTRANSMISION"/>
        <arg evaluator="xml" expression="$ctx:PROP_TICKETENVIO"/>
        <arg evaluator="xml" expression="$ctx:PROP_ANHOENVIO"/>
      </args>
    </payloadFactory>
    <property description="Content type" name="ContentType" scope="axis2" type="STRING"
value="application/json"/>
    <script description="Environment variables"
language="js"><![CDATA[mc.setProperty("PROP_KAFKA_BOOTSTRAP_SERVERS",
java.lang.System.getenv("KAFKA_BOOTSTRAP_SERVERS"));]]></script>
    <kafkaTransport.init>
      <bootstrapServers>{$ctx:PROP_KAFKA_BOOTSTRAP_SERVERS}</bootstrapServers>
    </kafkaTransport.init>
    <kafkaTransport.publishMessages>
      <topic>tpp_transmision_consulta</topic>
    </kafkaTransport.publishMessages>
    <drop description="End flow"/>
  </sequence>

```

Tabla 17. Código de flujo de proceso TransmisionSunatIRMErrorSeq

```

<sequence name="TransmisionSunatIRMErrorSeq" trace="disable" xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ TRANSMISION SUNAT OMA (ERROR SEQUENCE) +++++++"/>
    <property expression="get-property('ERROR_CODE')" name="ERROR_CODE"/>
    <property expression="get-property('ERROR_MESSAGE')" name="ERROR_MESSAGE"/>
    <property expression="get-property('ERROR_DETAIL')" name="ERROR_DETAIL"/>
    <property expression="get-property('ERROR_EXCEPTION')" name="ERROR_EXCEPTION"/>
  </log>
  <log description="Send retry">
    <property name="LOG" value="+++++++ REINTENTAR TRANSMISION SUNAT +++++++"/>
  </log>
  <dbreport description="Insert control">
    <connection>
      <pool>
        <driver>org.postgresql.Driver</driver>
        <url>${SYSTEM:DB_PG_URL}</url>
        <user>${SYSTEM:DB_PG_USR}</user>
        <password>${SYSTEM:DB_PG_PSW}</password>
      </pool>
    </connection>
    <statement>
      <sql><![CDATA[INSERT INTO tbl_sda_control (cpnnumerotransmision, cpccodtransmision, cpnid_evento,
cpcusuario_servicio) VALUES (?, ?, ?, ?)]]></sql>
      <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
      <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
      <parameter type="INTEGER" value="4"/>
      <parameter type="VARCHAR" value="SERV_TRANSMISION"/>
    </statement>
  </dbreport>
  <payloadFactory description="Create kafka payload" media-type="json">
    <format>
      {"numero_transmision":"$1", "codigo_transmision":"$2", "informacion_transmision":"$3",
"nro_reintento":"1"}</format>
    <args>
      <arg evaluator="xml" expression="$ctx:PROP_NUMTRANSMISION"/>
      <arg evaluator="xml" expression="$ctx:PROP_CODTRANSMISION"/>
      <arg evaluator="xml" expression="$ctx:infoTransmision"/>
    </args>
  </payloadFactory>
  <property description="Content type" name="ContentType" scope="axis2" type="STRING"
value="application/json"/>
  <script description="Environment variables"
language="js"><![CDATA[mc.setProperty("PROP_KAFKA_BOOTSTRAP_SERVERS",
java.lang.System.getenv("KAFKA_BOOTSTRAP_SERVERS"));]]></script>
  <kafkaTransport.init>
    <bootstrapServers>{ctx:PROP_KAFKA_BOOTSTRAP_SERVERS}</bootstrapServers>
  </kafkaTransport.init>
  <kafkaTransport.publishMessages>
    <topic>tpp_reintento_envio</topic>
  </kafkaTransport.publishMessages>
  <drop description="End flow"/>
</sequence>

```

Tabla 18. Código de flujo de proceso TransmisionSunatRSASeq

```

<sequence name="TransmisionSunatRSASeq" onError="TransmisionSunatRSAErrorSeq" trace="disable"
xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ TRANSMISION SUNAT RSA (SEQUENCE) +++++++"/>

```

```

    <property expression="$ctx:PROP_NUMTRANSMISION" name="NUMERO_TRANSMISION"/>
    <property expression="$ctx:PROP_CODTRANSMISION" name="CODIGO_TRANSMISION"/>
</log>
<payloadFactory description="Create payload" media-type="xml">
  <format>
    <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:wco:datamodel:pe:gob:sunat:controladuanero:ingreso:reconfisico:DeclaracionReconocimientoFisicoRequest:1" xmlns:urn1="urn:wco:datamodel:WCO:Declaration_DS:1">
      <soapenv:Body>
        $3
      </soapenv:Body>
    </soapenv:Envelope>
  </format>
  <args>
    <arg evaluator="xml" expression="$ctx:inlineXML"/>
  </args>
</payloadFactory>
<property description="Petición envío" expression="/" name="PROP_PETICIONENVIO" scope="default"
type="STRING"/>
<dbreport description="Update monitoreo">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <url>${SYSTEM:DB_PG_URL}</url>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[UPDATE tbl_sda_monitoreo SET cpcarchivoenvioxml = ?, cpcpeticionenvioxml = ? WHERE
cpnumerotransmision = ? AND cpcodtransmision = ?]]></sql>
    <parameter expression="$ctx:inlineXML" type="VARCHAR"/>
    <parameter expression="$ctx:PROP_PETICIONENVIO" type="VARCHAR"/>
    <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
    <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
  </statement>
</dbreport>
<property description="Content type" name="ContentType" scope="axis2" type="STRING" value="text/xml"/>
<call>
  <endpoint>
    <wsdl port="ReconocimientoFisicoPort" service="ReconocimientoFisicoService"
uri="https://ws.sunat.gob.pe/ws/controladuanero/ReconocimientoFisicoService.htm?wsdl">
      </wsdl>
    </endpoint>
  </call>
  <log description="Log response payload" level="full"/>
  <property description="Respuesta RSA" expression="/" name="PROP_RESPUESTARSA" scope="default"
type="STRING"/>
  <class description="Transform response" name="pe.com.tpp.transmisiones.TransformRSAResponse">
    <property expression="/" name="rsaResponse"/>
  </class>
  <log description="Log response RSA" level="custom">
    <property expression="$ctx:statusCode" name="statusCode"/>
    <property expression="$ctx:informacionMensaje" name="informacionMensaje"/>
    <property expression="$ctx:responseXML" name="responseXML"/>
  </log>
  <dbreport description="Update monitoreo">
    <connection>
      <pool>
        <driver>org.postgresql.Driver</driver>
        <url>${SYSTEM:DB_PG_URL}</url>
        <user>${SYSTEM:DB_PG_USR}</user>
        <password>${SYSTEM:DB_PG_PSW}</password>
      </pool>
    </connection>
  </dbreport>
</call>
</log>

```

```

    </pool>
  </connection>
  <statement>
    <sql><![CDATA[UPDATE tbl_sda_monitoreo SET cpcrespuestaxml = ?, cpncodrespuesta = ?,
cpcinfomensajerespuesta = ?, cpdfecharespuesta = CURRENT_TIMESTAMP WHERE cpnnumerotransmision = ?
AND cpccodtransmision = ?]]></sql>
    <parameter expression="/" type="VARCHAR"/>
    <parameter expression="$ctx:statusCode" type="INTEGER"/>
    <parameter expression="$ctx:informacionMensaje" type="VARCHAR"/>
    <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
    <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
  </statement>
</dbreport>
<switch source="$ctx:statusCode">
  <case regex="1">
    <log description="Case 1">
      <property name="LOG" value="+++++++ STATUS-CODE: 1 +++++++"/>
    </log>
    <property description="Evento 13" name="PROP_EVENTOID" scope="default" type="STRING"
value="13"/>
  </case>
  <default>
    <log description="Default">
      <property name="LOG" value="+++++++ STATUS-CODE:8 +++++++"/>
      <property expression="$ctx:statusCode" name="STATUS-CODE"/>
    </log>
    <property description="Evento 9" name="PROP_EVENTOID" scope="default" type="STRING" value="9"/>
  </default>
</switch>
<dbreport description="Insert control">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <url>${SYSTEM:DB_PG_URL}</url>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[INSERT INTO tbl_sda_control (cpnnumerotransmision, cpccodtransmision, cpnid_evento,
cpncodrespuesta, cpcinfomensajerespuesta, cpcrespuestaxml, cpcusuario_servicio) VALUES (?, ?, ?, ?, ?,
?)]></sql>
    <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
    <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
    <parameter expression="$ctx:PROP_EVENTOID" type="INTEGER"/>
    <parameter expression="$ctx:statusCode" type="INTEGER"/>
    <parameter expression="$ctx:informacionMensaje" type="VARCHAR"/>
    <parameter expression="$ctx:responseXML" type="VARCHAR"/>
    <parameter type="VARCHAR" value="SERV_TRANSMISION"/>
  </statement>
</dbreport>
<log description="Send response to on-premise">
  <property name="LOG" value="+++++++ ENVIAR RESPUESTA DE RSA +++++++"/>
</log>
<dblookup description="Select data transmision">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <url>${SYSTEM:DB_PG_URL}</url>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>

```

```

<statement>
  <sql><![CDATA[SELECT cpnid_tticket,cpnid_envioaduana FROM tbl_sda_rsa WHERE cpnid_rsa = ? AND
  cpccodtransmision = ?]]></sql>
  <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
  <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
  <result column="cpnid_tticket" name="cpnid_tticket"/>
  <result column="cpnid_envioaduana" name="cpnid_envioaduana"/>
</statement>
</dblookup>
<payloadFactory description="Enviar respuesta" media-type="json">
  <format>
    {"numero_transmision": "$1", "codigo_transmision": "$2", "codigo_respuesta": "$3",
    "mensaje_respuesta": "$4", "id_ticket": "$5", "id_envioaduana": "$6", "respuesta_xml": "$7"}</format>
  <args>
    <arg evaluator="xml" expression="$ctx:PROP_NUMTRANSMISION"/>
    <arg evaluator="xml" expression="$ctx:PROP_CODTRANSMISION"/>
    <arg evaluator="xml" expression="$ctx:statusCode"/>
    <arg evaluator="xml" expression="$ctx:informacionMensaje"/>
    <arg evaluator="xml" expression="$ctx:cpnid_tticket"/>
    <arg evaluator="xml" expression="$ctx:cpnid_envioaduana"/>
    <arg evaluator="xml" expression="$ctx:responseXMLBase64"/>
  </args>
</payloadFactory>
<log description="Log payload" level="full"/>
<property description="Content type" name="ContentType" scope="axis2" type="STRING"
value="application/json"/>
<script description="Environment variables"
language="js"><![CDATA[mc.setProperty("PROP_KAFKA_BOOTSTRAP_SERVERS",
java.lang.System.getenv("KAFKA_BOOTSTRAP_SERVERS"));]]></script>
<kafkaTransport.init>
  <bootstrapServers>{$ctx:PROP_KAFKA_BOOTSTRAP_SERVERS}</bootstrapServers>
</kafkaTransport.init>
<kafkaTransport.publishMessages>
  <topic>tpp_transmision_resultado</topic>
</kafkaTransport.publishMessages>
<log description="Log end">
  <property name="LOG" value="+++++++ FIN DE TRANSMISION ++++"/>
</log>
<drop description="End transmision"/>
</sequence>

```

Tabla 19. Código de flujo de proceso TransmisionSunatRSAErrorSeq

```

<sequence name="TransmisionSunatRSAErrorSeq" trace="disable" xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ TRANSMISION SUNAT RSA (ERROR SEQUENCE) ++++"/>
    <property expression="get-property('ERROR_CODE')" name="ERROR_CODE"/>
    <property expression="get-property('ERROR_MESSAGE')" name="ERROR_MESSAGE"/>
    <property expression="get-property('ERROR_DETAIL')" name="ERROR_DETAIL"/>
    <property expression="get-property('ERROR_EXCEPTION')" name="ERROR_EXCEPTION"/>
  </log>
  <log description="Send retry">
    <property name="LOG" value="+++++++ REINTENTAR TRANSMISION SUNAT RSA ++++"/>
  </log>
  <dbreport description="Insert control">
    <connection>
      <pool>
        <driver>org.postgresql.Driver</driver>
        <uri>${SYSTEM:DB_PG_URL}</uri>
        <user>${SYSTEM:DB_PG_USR}</user>
      </pool>
    </connection>
  </dbreport>

```

```

    <password>$$SYSTEM:DB_PG_PSW</password>
  </pool>
</connection>
<statement>
  <sql><![CDATA[INSERT INTO tbl_sda_control (cpnnumerotransmission, cpccodtransmission, cpnid_evento,
cpcusuario_servicio) VALUES (?, ?, ?, ?)]]></sql>
  <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
  <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
  <parameter type="INTEGER" value="4"/>
  <parameter type="VARCHAR" value="SERV_TRANSMISION"/>
</statement>
</dbreport>
<payloadFactory description="Create kafka payload" media-type="json">
  <format>
    {"numero_transmission":"$1", "codigo_transmission":"$2", "informacion_transmission":"$3",
"nro_reintento":"1"}</format>
  <args>
    <arg evaluator="xml" expression="$ctx:PROP_NUMTRANSMISION"/>
    <arg evaluator="xml" expression="$ctx:PROP_CODTRANSMISION"/>
    <arg evaluator="xml" expression="$ctx:infoTransmission"/>
  </args>
</payloadFactory>
<property description="Content type" name="ContentType" scope="axis2" type="STRING"
value="application/json"/>
<script description="Environment variables"
language="js"><![CDATA[mc.setProperty("PROP_KAFKA_BOOTSTRAP_SERVERS",
java.lang.System.getenv("KAFKA_BOOTSTRAP_SERVERS"));]]></script>
  <kafkaTransport.init>
    <bootstrapServers>{$ctx:PROP_KAFKA_BOOTSTRAP_SERVERS}</bootstrapServers>
  </kafkaTransport.init>
  <kafkaTransport.publishMessages>
    <topic>tpp_reintento_envio</topic>
  </kafkaTransport.publishMessages>
  <drop description="End flow"/>
</sequence>

```

-Desarrollo de Componente Consulta

-Entrada de evento (Inbound-Endpoints)

Se crea la entrada de evento "ConsultaInboundEP.xml" que leerá las entradas (Formato JSON) enviados a la cola de mensajes "tpp_transmision_consulta". Por cada lectura exitosa de documento seguirá el flujo de proceso "ConsultaMainSeq" (Sequence). En caso de lectura errónea seguirá el flujo de proceso "ConsultaMainErrorSeq" (Sequence).

Se define los parámetros de configuración para la entrada de evento "ConsultaInboundEP.xml".

- Sequential: Lectura secuencial de los mensajes de entrada. Valor configurado: true
- Interval: Intervalo de lectura de los mensajes de entrada en milisegundos. Valor configurado: 5000
- Coordination: Ejecución en modo cluster (false) o en un solo nodo (true). Valor configurado: true
- Inbound.behavior: Comportamiento de lectura de datos. Valor configurado: pooling
- Topic.name: Nombre de la cola de mensajes de entrada para la lectura. Valor configurado: tpp_transmision_consulta

- Poll.timeout: Tiempo máximo de espera de lectura en milisegundos. Valor configurado: 1000
- Max.poll.records: Cantidad máxima de mensajes a lectura de forma periódica. Valor configurado: 5
- ContentType: Formato de mensajes de entrada. Valor configurado: application/json

Tabla 20. Código de entrada de evento ConsultaInboundEP

```
<inboundEndpoint class="org.wso2.carbon.inbound.kafka.KafkaMessageConsumer" name="ConsultaInboundEP"
onError="ConsultaMainErrorSeq" sequence="ConsultaMainSeq" suspend="false"
xmlns="http://ws.apache.org/ns/synapse">
  <parameters>
    <parameter name="sequential">true</parameter>
    <parameter name="interval">5000</parameter>
    <parameter name="coordination">true</parameter>
    <parameter name="inbound.behavior">polling</parameter>
    <parameter name="topic.name">tpp_transmission_consulta</parameter>
    <parameter name="poll.timeout">1000</parameter>
    <parameter name="max.poll.records">5</parameter>
    <parameter name="bootstrap.servers">${SYSTEM:KAFKA_BOOTSTRAP_SERVERS}</parameter>
    <parameter name="group.id">consultawso2service</parameter>
    <parameter name="contentType">application/json</parameter>
  </parameters>
</inboundEndpoint>
```

- Flujos de proceso (Sequence)

Se desarrollan dos flujos de procesos principales y 2 flujos secundarios:

ConsultaMainSeq: Contiene la lógica principal del flujo de proceso Consulta:

- Validación del límite de tiempo de consulta según configuración.
- Registros en tablas de eventos y control del proceso.
- Continuidad al subproceso "ConsultaSunatSeq".

ConsultaMainErrorSeq: Contiene la lógica de control de errores del flujo de proceso "ConsultaMainSeq":

- Registro de logs con el código y mensaje de error.
- Ejecuta el fin del flujo de proceso "ConsultaMainSeq".

ConsultaSunatSeq: Contiene la lógica principal del flujo de proceso:

- Creación de cliente de comunicación para consultar el estado de transmisión al Web Services de SUNAT.
- Lectura de respuesta (Codigo y mensaje de respuesta) de Web Services de SUNAT.
- Registros en tablas de eventos y control del proceso.

ConsultaSunatErrorSeq: Contiene la lógica de control de errores del flujo de proceso "ConsultaSunatSeq":

- Registro de logs con el código y mensaje de error.
- Ejecuta el fin del flujo de proceso "ConsultaMainSeq".

Tabla 21. Código de flujo de proceso ConsultaMainSeq

```

<sequence name="ConsultaMainSeq" onError="ConsultaMainErrorSeq" trace="disable"
xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log properties" level="custom">
    <property name="LOG" value="+++++++ CONSULTA (MAIN SEQUENCE) +++++++"/>
    <property expression="json-eval($)" name="MENSAJE_JSON"/>
  </log>
  <propertyGroup description="Properties transmission">
    <property expression="//numero_transmission" name="PROP_NUMTRANSMISION" scope="default"
type="STRING"/>
    <property expression="//codigo_transmission" name="PROP_CODTRANSMISION" scope="default"
type="STRING"/>
    <property expression="//ticket_envio" name="PROP_TICKETENVIO" scope="default" type="STRING"/>
    <property expression="//anho_envio" name="PROP_ANHOENVIO" scope="default" type="STRING"/>
  </propertyGroup>
  <dbreport description="Insert control">
    <connection>
      <pool>
        <driver>org.postgresql.Driver</driver>
        <url>${SYSTEM:DB_PG_URL}</url>
        <user>${SYSTEM:DB_PG_USR}</user>
        <password>${SYSTEM:DB_PG_PSW}</password>
      </pool>
    </connection>
    <statement>
      <sql><![CDATA[INSERT INTO tbl_sda_control (cpnnumerotransmission, cpccodtransmission, cpnid_evento,
cpcusuario_servicio) VALUES (?, ?, ?, ?)]]></sql>
      <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
      <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
      <parameter type="INTEGER" value="3"/>
      <parameter type="VARCHAR" value="SERV_CONSULTA"/>
    </statement>
  </dbreport>
  <dblookup description="Select config">
    <connection>
      <pool>
        <driver>org.postgresql.Driver</driver>
        <url>${SYSTEM:DB_PG_URL}</url>
        <user>${SYSTEM:DB_PG_USR}</user>
        <password>${SYSTEM:DB_PG_PSW}</password>
      </pool>
    </connection>
    <statement>
      <sql><![CDATA[SELECT cpcusuariows, cpclavews FROM tbl_sda_configuracion WHERE cpccodtransmission
= ?]]></sql>
      <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
      <result column="cpclavews" name="cpclavews"/>
      <result column="cpcusuariows" name="cpcusuariows"/>
    </statement>
  </dblookup>
  <propertyGroup description="Properties config">
    <property expression="$ctx:cpcusuariows" name="PROP_USUARIOWS" scope="default" type="STRING"/>
    <property expression="$ctx:cpclavews" name="PROP_CLAVEWS" scope="default" type="STRING"/>
  </propertyGroup>
  <dblookup description="Select ID envio aduana">
    <connection>
      <pool>
        <driver>org.postgresql.Driver</driver>
        <url>${SYSTEM:DB_PG_URL}</url>
        <user>${SYSTEM:DB_PG_USR}</user>
        <password>${SYSTEM:DB_PG_PSW}</password>
      </pool>
    </connection>
  </dblookup>

```



```

</connection>
<statement>
  <sql><![CDATA[SELECT COALESCE(cpnid_envioaduana::varchar, '') AS cpnid_envioaduana FROM
tbl_sda_monitoreo WHERE cpnnumerotransmision = ? AND cpccodtransmision = ?]]></sql>
  <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
  <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
  <result column="cpnid_envioaduana" name="cpnid_envioaduana"/>
</statement>
</dblookup>
<propertyGroup description="Transmision data">
  <property expression="$ctx:cpnid_envioaduana" name="PROP_IDENVIOADUANA" scope="default"
type="STRING"/>
  <property name="PROP_IDTICKET" scope="default" type="STRING" value=""/>
</propertyGroup>
<switch source="$ctx:PROP_CODTRANSMISION">
  <case regex="0121|0122">
    <log description="IRM">
      <property name="LOG" value="+++++++ IRM ++++"/>
    </log>
    <propertyGroup description="Tipo">
      <property name="PROP_TIPOTRANSMISION" scope="default" type="STRING" value="IRM"/>
    </propertyGroup>
    <sequence key="ConsultaSunatSeq"/>
    <drop description="End flow"/>
  </case>
  <case regex="RSA">
    <log description="RSA">
      <property name="LOG" value="+++++++ IRM ++++"/>
    </log>
    <propertyGroup description="Tipo">
      <property name="PROP_TIPOTRANSMISION" scope="default" type="STRING" value="IRM"/>
    </propertyGroup>
    <sequence key="ConsultaSunatSeq"/>
    <drop description="End flow"/>
  </case>
  <default>
    <drop description="End flow"/>
  </default>
</switch>
</sequence>

```

Tabla 22. Código de flujo de proceso ConsultaMainErrorSeq

```

<sequence name="ConsultaMainErrorSeq" trace="disable" xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ CONSULTA (MAIN ERROR SEQUENCE) ++++"/>
    <property expression="get-property('ERROR_CODE')" name="ERROR_CODE"/>
    <property expression="get-property('ERROR_MESSAGE')" name="ERROR_MESSAGE"/>
    <property expression="get-property('ERROR_DETAIL')" name="ERROR_DETAIL"/>
    <property expression="get-property('ERROR_EXCEPTION')" name="ERROR_EXCEPTION"/>
  </log>
  <drop description="End flow"/>
</sequence>

```

Tabla 23. Código de flujo de proceso ConsultaSunatSeq

```

<sequence name="ConsultaSunatSeq" onError="ConsultaSunatErrorSeq" trace="disable"
xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log properties">

```

```

<property name="LOG" value="+++++++ CONSULTA SUNAT (SEQUENCE) +++++++"/>
<property expression="$ctx:PROP_TIPOTRANSMISION" name="TIPO_TRANSMISION"/>
<property expression="$ctx:PROP_NUMTRANSMISION" name="NUMERO_TRANSMISION"/>
<property expression="$ctx:PROP_CODTRANSMISION" name="CODIGO_TRANSMISION"/>
<property expression="$ctx:PROP_TICKETENVIO" name="TICKET_ENVIO"/>
<property expression="$ctx:PROP_ANHOENVIO" name="ANHO_ENVIO"/>
<property expression="$ctx:PROP_IDTICKET" name="ID_TICKET"/>
<property expression="$ctx:PROP_IDENVIOADUANA" name="ID_ENVIOADUANA"/>
</log>
<payloadFactory description="Create payload" media-type="xml">
  <format>
    <soapenv:Envelope xmlns:ser="http://services.sigad.sunat.gob.pe"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
      <soapenv:Body>
        <realizarConsulta xmlns="http://services.sigad.sunat.gob.pe">
          <parametrosConsulta
xmlns=""><![CDATA[<consulta><tipo>1</tipo><parametros><numeroTicket>$3</numeroTicket><annoTicket>$4
</annoTicket></parametros></consulta>]]></parametrosConsulta>
          </realizarConsulta>
        </soapenv:Body>
      </soapenv:Envelope>
    </format>
    <args>
      <arg evaluator="xml" expression="$ctx:PROP_USUARIOS"/>
      <arg evaluator="xml" expression="$ctx:PROP_CLAVES"/>
      <arg evaluator="xml" expression="$ctx:PROP_TICKETENVIO"/>
      <arg evaluator="xml" expression="$ctx:PROP_ANHOENVIO"/>
    </args>
  </payloadFactory>
  <log description="Log request payload" level="full"/>
  <property description="Petición consulta" expression="/" name="PROP_PETICIONCONSULTA"
scope="default" type="STRING"/>
  <dbreport description="Update monitoreo">
    <connection>
      <pool>
        <driver>org.postgresql.Driver</driver>
        <url>${SYSTEM:DB_PG_URL}</url>
        <user>${SYSTEM:DB_PG_USR}</user>
        <password>${SYSTEM:DB_PG_PSW}</password>
      </pool>
    </connection>
    <statement>
      <sql><![CDATA[UPDATE tbl_sda_monitoreo SET cpcpcionconsultaxml = ? WHERE
cpnnumerotransmision = ? AND cpccodtransmision = ? AND cpcicketenvio = ?]]></sql>
      <parameter expression="$ctx:PROP_PETICIONCONSULTA" type="VARCHAR"/>
      <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
      <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
      <parameter expression="$ctx:PROP_TICKETENVIO" type="VARCHAR"/>
    </statement>
  </dbreport>
  <property description="Content type" name="ContentType" scope="axis2" type="STRING" value="text/xml"/>
  <call>
    <endpoint>
      <wsdl port="ReceptorWebServiceServiceImplPort" service="ReceptorService.htm"
uri="${SYSTEM:WSDL_URI}">
        </wsdl>
      </endpoint>
    </call>
    <log description="Log response payload" level="full"/>
    <propertyGroup description="Respuesta consulta">
      <property expression="/" name="PROP_RESPUESTACONSULTA" scope="default" type="STRING"/>
      <property expression="/*[local-name()='realizarConsultaResultado']/text()"
name="PROP_RESPUESTAXMLBASE64" scope="default" type="STRING"/>

```

```

</propertyGroup>
<class description="Transform response" name="pe.com.tpp.mediators.consulta.TransformResponse">
  <property expression="$ctx:PROP_RESPUESTAXMLBASE64" name="encodedResponse"/>
</class>
<dbreport description="Update monitoreo">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <url>${SYSTEM:DB_PG_URL}</url>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[UPDATE tbl_sda_monitoreo SET cpcrespuestaconsultaxml = ?, cpncodrespuesta = ?,
cpcinformajerespuesta = ?, cpdfcharespuesta = CURRENT_TIMESTAMP, cpcrespuestaxml = ? WHERE
cpnnumero transmision = ? AND cpccodtransmision = ? AND cpcticketenvio = ?]]></sql>
    <parameter expression="$ctx:PROP_RESPUESTACONSULTA" type="VARCHAR"/>
    <parameter expression="$ctx:statusCode" type="INTEGER"/>
    <parameter expression="$ctx:informacionMensaje" type="VARCHAR"/>
    <parameter expression="$ctx:responseXML" type="VARCHAR"/>
    <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
    <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
    <parameter expression="$ctx:PROP_TICKETENVIO" type="VARCHAR"/>
  </statement>
</dbreport>
<switch source="$ctx:PROP_EVENTOID">
  <case regex="13/9/11">
    <log description="Case 13/9/11">
      <property name="LOG" value="+++++++ EVENTO-ID: 13/9/11 +++++++"/>
    </log>
  </case>
  <case regex="7">
    <log description="Case 7">
      <property name="LOG" value="+++++++ EVENTO-ID: 7 +++++++"/>
    </log>
    <dbreport description="Insert control">
      <connection>
        <pool>
          <driver>org.postgresql.Driver</driver>
          <url>${SYSTEM:DB_PG_URL}</url>
          <user>${SYSTEM:DB_PG_USR}</user>
          <password>${SYSTEM:DB_PG_PSW}</password>
        </pool>
      </connection>
      <statement>
        <sql><![CDATA[INSERT INTO tbl_sda_control (cpnnumero transmision, cpccodtransmision,
cpnid_evento, cpncodrespuesta, cpcinformajerespuesta, cpcrespuestaxml, cpcusuario_servicio) VALUES (?, ?,
?, ?, ?, ?, ?)]]></sql>
        <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
        <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
        <parameter expression="$ctx:PROP_EVENTOID" type="INTEGER"/>
        <parameter expression="$ctx:statusCode" type="INTEGER"/>
        <parameter expression="$ctx:informacionMensaje" type="VARCHAR"/>
        <parameter expression="$ctx:responseXML" type="VARCHAR"/>
        <parameter type="VARCHAR" value="SERV_CONSULTA"/>
      </statement>
    </dbreport>
    <drop description="End flow"/>
  </case>
  <default>
    <log description="Default">
      <property name="LOG" value="+++++++ EVENTO-ID: 10 (REINTENTAR CONSULTA) +++++++"/>
    </log>
  </default>
</switch>

```

```

</log>
<dbreport description="Insert control">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <url>${SYSTEM:DB_PG_URL}</url>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[INSERT INTO tbl_sda_control (cpnnumerotransmision, cpccodtransmision,
cpnid_evento, cpncodrespuesta, cpcinfolmensajerespuesta, cpcrespuestaxml, cpcusuario_servicio) VALUES (?, ?,
?, ?, ?, ?, ?)]]></sql>
    <parameter expression="${Ctx:PROP_NUMTRANSMISION}" type="INTEGER"/>
    <parameter expression="${Ctx:PROP_CODTRANSMISION}" type="CHAR"/>
    <parameter expression="${Ctx:PROP_EVENTOID}" type="INTEGER"/>
    <parameter expression="${Ctx:statusCode}" type="INTEGER"/>
    <parameter expression="${Ctx:informacionMensaje}" type="VARCHAR"/>
    <parameter expression="${Ctx:responseXML}" type="VARCHAR"/>
    <parameter type="VARCHAR" value="SERV_CONSULTA"/>
  </statement>
</dbreport>
<sequence key="ConsultaSunatOMAErrorSeq"/>
<drop description="End flow"/>
</default>
</switch>
<dbreport description="Insert control">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <url>${SYSTEM:DB_PG_URL}</url>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[INSERT INTO tbl_sda_control (cpnnumerotransmision, cpccodtransmision, cpnid_evento,
cpncodrespuesta, cpcinfolmensajerespuesta, cpcrespuestaxml, cpcusuario_servicio) VALUES (?, ?, ?, ?, ?,
?))]></sql>
    <parameter expression="${Ctx:PROP_NUMTRANSMISION}" type="INTEGER"/>
    <parameter expression="${Ctx:PROP_CODTRANSMISION}" type="CHAR"/>
    <parameter expression="${Ctx:PROP_EVENTOID}" type="INTEGER"/>
    <parameter expression="${Ctx:statusCode}" type="INTEGER"/>
    <parameter expression="${Ctx:informacionMensaje}" type="VARCHAR"/>
    <parameter expression="${Ctx:responseXML}" type="VARCHAR"/>
    <parameter type="VARCHAR" value="SERV_CONSULTA"/>
  </statement>
</dbreport>
<dblookup description="Actualizar minutos">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <url>${SYSTEM:DB_PG_URL}</url>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[SELECT funsda_actualizarminutos(?, ?, ?)]]></sql>
    <parameter expression="${Ctx:PROP_NUMTRANSMISION}" type="INTEGER"/>
    <parameter expression="${Ctx:PROP_CODTRANSMISION}" type="VARCHAR"/>
    <parameter expression="${Ctx:PROP_TIPOTRANSMISION}" type="VARCHAR"/>

```

```

    </statement>
  </dblookup>
  <log description="Send response to on-premise">
    <property name="LOG" value="+++++++ ENVIAR RESPUESTA DE SUNAT A ON-PREMISE +++++++"/>
  </log>
  <payloadFactory description="Enviar respuesta" media-type="json">
    <format>
      {"numero_transmision":"$1", "codigo_transmision":"$2", "codigo_respuesta":"$3",
"mensaje_respuesta":"$4", "id_ticket":"$5", "id_envioaduana":"$6", "respuesta_xml":"$7"}</format>
    <args>
      <arg evaluator="xml" expression="$ctx:PROP_NUMTRANSMISION"/>
      <arg evaluator="xml" expression="$ctx:PROP_CODTRANSMISION"/>
      <arg evaluator="xml" expression="$ctx:statusCode"/>
      <arg evaluator="xml" expression="$ctx:informacionMensaje"/>
      <arg evaluator="xml" expression="$ctx:PROP_IDTICKET"/>
      <arg evaluator="xml" expression="$ctx:PROP_IDENVIADUANA"/>
      <arg evaluator="xml" expression="$ctx:PROP_RESPUESTAXMLBASE64"/>
    </args>
  </payloadFactory>
  <log description="Log json payload" level="full"/>
  <property description="Content type" name="ContentType" scope="axis2" type="STRING"
value="application/json"/>
  <script description="Environment variables"
language="js"><![CDATA[mc.setProperty("PROP_KAFKA_BOOTSTRAP_SERVERS",
java.lang.System.getenv("KAFKA_BOOTSTRAP_SERVERS"));]]></script>
  <kafkaTransport.init>
    <bootstrapServers>{$ctx:PROP_KAFKA_BOOTSTRAP_SERVERS}</bootstrapServers>
  </kafkaTransport.init>
  <kafkaTransport.publishMessages>
    <topic>tpp_transmision_resultado</topic>
  </kafkaTransport.publishMessages>
  <log description="Log end">
    <property name="LOG" value="+++++++ FIN DE TRANSMISION +++++++"/>
  </log>
  <drop description="End transmision"/>
</sequence>

```

Tabla 24. Código de flujo de proceso ConsultaSunatErrorSeq

```

<?xml version="1.0" encoding="UTF-8"?>
<sequence name=" ConsultaSunatErrorSeq" trace="disable" xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ CONSULTA (MAIN ERROR SEQUENCE) +++++++"/>
    <property expression="get-property('ERROR_CODE')" name="ERROR_CODE"/>
    <property expression="get-property('ERROR_MESSAGE')" name="ERROR_MESSAGE"/>
    <property expression="get-property('ERROR_DETAIL')" name="ERROR_DETAIL"/>
    <property expression="get-property('ERROR_EXCEPTION')" name="ERROR_EXCEPTION"/>
  </log>
  <drop description="End flow"/>
</sequence>

```

-Desarrollo de Componente Reintento Transmisión

-Entrada de evento (Inbound-Endpoints)

Se crearán las siguientes entradas de evento.

- “ReintentoEnvioInboundEP.xml” que leerá las entradas (Formato JSON) enviados a la cola de mensajes “tpp_transmision_envio”. Por cada lectura exitosa de documento seguirá el flujo de proceso “ReintentoEnvioMainSeq” (Sequence). En caso de lectura errónea seguirá el flujo de proceso “ReintentoEnvioMainErrorSeq” (Sequence).

Se define los parámetros de configuración para la entrada de evento “ReintentoEnvioInboundEP.xml”.

- Sequential: Lectura secuencial de los mensajes de entrada. Valor configurado: true
- Interval: Intervalo de lectura de los mensajes de entrada en milisegundos. Valor configurado: 5000
- Coordination: Ejecución en modo cluster (false) o en un solo nodo (true). Valor configurado: true
- Inbound.behavior: Comportamiento de lectura de datos. Valor configurado: pooling
- Topic.name: Nombre de la cola de mensajes de entrada para la lectura de transmisiones. Valor configurado: tpp_reintento_envio
- Poll.timeout: Tiempo máximo de espera de lectura en milisegundos. Valor configurado: 1000
- Max.poll.records: Cantidad máxima de mensajes a lectura de forma periódica. Valor configurado: 5
- ContentType: Formato de mensajes de entrada. Valor configurado: application/json

Tabla 25. Código de entrada de evento ReintentoEnvioInboundEP

```
<inboundEndpoint class="org.wso2.carbon.inbound.kafka.KafkaMessageConsumer"
name="ReintentoEnvioInboundEP" onError="ReintentoEnvioMainErrorSeq" sequence="ReintentoEnvioMainSeq"
suspend="false" xmlns="http://ws.apache.org/ns/synapse">
  <parameters>
    <parameter name="sequential">true</parameter>
    <parameter name="interval">5000</parameter>
    <parameter name="coordination">true</parameter>
    <parameter name="inbound.behavior">polling</parameter>
    <parameter name="topic.name">tpp_reintento_envio</parameter>
    <parameter name="poll.timeout">1000</parameter>
    <parameter name="max.poll.records">5</parameter>
    <parameter name="bootstrap.servers">${SYSTEM:KAFKA_BOOTSTRAP_SERVERS}</parameter>
    <parameter name="group.id">reintentostransmisionwso2service</parameter>
    <parameter name="contentType">application/json</parameter>
  </parameters>
</inboundEndpoint>
```

- “ReintentoEnvioDBEventInboundEP.xml” que leerá las entradas (Formato JSON) desde la tabla de reintendo de envíos pendientes “tbl_sda_reintento_envio”. Por cada lectura exitosa de documento seguirá el flujo de proceso “ReintentoEnvioMainSeq” (Sequence). En caso de lectura errónea seguirá el flujo de proceso “ReintentoEnvioMainErrorSeq” (Sequence).

Se define los parámetros de configuración para la entrada de evento “ReintentoEnvioDBEventInboundEP.xml”.

- Sequential: Lectura secuencial de los registros de las tablas. Valor configurado: true

- Interval: Intervalo de lectura de los mensajes de entrada en milisegundos. Valor configurado: 300000
- Coordination: Ejecución en modo cluster (false) o en un solo nodo (true). Valor configurado: true
- Inbound.behavior: Comportamiento de lectura de datos. Valor configurado: pooling
- driverName: Nombre del driver de conexión a la Base de Datos. Valor configurado: org.postgresql.Driver
- url: Cadena de conexión a Base de Datos. Valor configurado: \$SYSTEM:DB_PG_URL
- username: Usuario de conexión a Base de Datos. Valor configurado: \$SYSTEM:DB_PG_USR
- password: Password de conexión a Base de Datos. Valor configurado: \$SYSTEM:DB_PG_PSW
- tableName: Tabla de consulta de transmisiones pendientes de reintentar. Valor configurado: tbl_sda_reintento_envio
- filteringCriteria: Criterio de filtro de registros. Valor configurado: byBooleanColumn
- cpcestadoenvio: Columna que aplica el criterio de filtro de registros. Valor configurado: cpcestadoenvio
- primaryKey: Llave primaria de la tabla. Valor configurado: cpnid_reintentoenvio

Tabla 26. Código de entrada de evento ReintentoEnvioDBEventInboundEP

```
<inboundEndpoint class="org.wso2.carbon.inbound.poll.dbeventlistener.DBEventPollingConsumer"
name="ReintentoEnvioDBEventInboundEP" onError="ReintentoEnvioMainErrorSeq"
sequence="ReintentoEnvioMainSeq" suspend="false" xmlns="http://ws.apache.org/ns/synapse">
  <parameters>
    <parameter name="interval">300000</parameter>
    <parameter
name="class">org.wso2.carbon.inbound.poll.dbeventlistener.DBEventPollingConsumer</parameter>
    <parameter name="sequential">true</parameter>
    <parameter name="coordination">true</parameter>
    <parameter name="inbound.behavior">polling</parameter>
    <parameter name="driverName">org.postgresql.Driver</parameter>
    <parameter name="url">$SYSTEM:DB_PG_URL</parameter>
    <parameter name="username">$SYSTEM:DB_PG_USR</parameter>
    <parameter name="password">$SYSTEM:DB_PG_PSW</parameter>
    <parameter name="tableName">tbl_sda_reintento_envio</parameter>
    <parameter name="filteringCriteria">byBooleanColumn</parameter>
    <parameter name="filteringColumnName">cpcestadoenvio</parameter>
    <parameter name="primaryKey">cpnid_reintentoenvio</parameter>
    <parameter name="connectionValidationQuery">SELECT 1</parameter>
  </parameters>
</inboundEndpoint>
```

- Flujos de proceso (Sequence)

Se desarrollan dos flujos de procesos principales y 2 flujos secundarios:

ReintentoEnvioMainSeq: Contiene la lógica principal del flujo de proceso Consulta:

- Validación del límite de tiempo de reintento de envío según configuración.
- Registros en tablas de eventos y control del proceso.

- Switch de decisión: Si es tipo IRM delegar la continuidad al subproceso "ReintentoEnvioSunatIRMSeq.xml". Si es tipo RSA delegar la continuidad al subproceso "ReintentoEnvioSunatRSASeq.xml".
- Notifica por correo cuando el reintento de envío de una transmisión supera el límite de tiempo.

ReintentoEnvioMainErrorSeq: Contiene la lógica de control de errores del flujo de proceso "ReintentoEnvioMainSeq":

- Registro de logs con el código y mensaje de error.
- Ejecuta el fin del flujo de proceso "ReintentoEnvioMainSeq".

ReintentoEnvioSunatIRMSeq: Contiene la lógica principal del flujo de proceso:

- Creación de cliente de comunicación para reintentar el envío del documento de transmisión a Web Services IRM de SUNAT.
- Lectura de respuesta (Fecha de recepción, Nro de Ticket) de Web Services de SUNAT.
- Registros en tablas de eventos y control del proceso.

ReintentoEnvioSunatIRMErrorSeq: Contiene la lógica de control de errores del flujo de proceso "ReintentoEnvioSunatIRMSeq":

- Registro de logs con el código y mensaje de error.
- Ejecuta el fin del flujo de proceso "ReintentoEnvioSunatIRMSeq".

ReintentoEnvioSunatRSASeq: Contiene la lógica principal del flujo de proceso para RSA:

- Creación de cliente de comunicación para reintentar el envío del documento de transmisión a Web Services RSA de SUNAT.
- Lectura de respuesta (Estado de transmisión) de Web Services de SUNAT.
- Registros en tablas de eventos y control del proceso.

ReintentoEnvioSunatRSAErrorSeq: Contiene la lógica de control de errores del flujo de proceso "ReintentoEnvioSunatRSASeq":

- Registro de logs con el código y mensaje de error.
- Ejecutar el fin del flujo de proceso "ReintentoEnvioSunatRSASeq".

Tabla 27. Código de flujo de proceso ReintentoEnvioMainSeq

```
<sequence name="ReintentoEnvioMainSeq" onError="ReintentoEnvioMainErrorSeq" trace="disable"
xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ REINTENTO ENVIO (MAIN SEQUENCE) +++++++"/>
    <property expression="json-eval($)" name="MENSAJE_JSON"/>
  </log>
  <propertyGroup description="Transmision properties">
    <property expression="//numero_transmision" name="PROP_NUMTRANSMISION" scope="default"
type="STRING"/>
    <property expression="//codigo_transmision" name="PROP_CODTRANSMISION" scope="default"
type="STRING"/>
  </propertyGroup>
</sequence>
```



```

    <property expression="//informacion_transmision" name="PROP_INFOTRANSMISION" scope="default"
type="STRING"/>
    <property expression="//nro_reintento" name="PROP_NUMREINTENTO" scope="default" type="STRING"/>
    <property expression="//cpdfechasgtereintento" name="PROP_FECHASGTEREINTENTO" scope="default"
type="STRING"/>
</propertyGroup>
<script description="Environment variables" language="js"><![CDATA[mc.setProperty("PROP_DB_PG_URL",
java.lang.System.getenv("DB_PG_URL"));mc.setProperty("PROP_DB_PG_USERNAME",
java.lang.System.getenv("DB_PG_USERNAME"));mc.setProperty("PROP_DB_PG_PASSWORD",
java.lang.System.getenv("DB_PG_PASSWORD"));]]></script>
<class description="Validar Reintento fecha"
name="pe.com.tpp.mediators.reintentotransmision.ValidarReintentoFecha">
    <property expression="$ctx:PROP_NUMREINTENTO" name="nro_reintento"/>
    <property expression="$ctx:PROP_FECHASGTEREINTENTO" name="fecha_sgte_reintento"/>
</class>
<switch source="$ctx:flag_reintento_cola">
    <case regex="SI">
        <log description="Continuar">
            <property name="LOG" value="+++++++ CONTINUAR CON REINTENTO ENVIO ++++++"/>
        </log>
    </case>
    <default>
        <log description="Validar fecha de reintento">
            <property name="LOG" value="+++++++ EVALUAR FECHA REINTENTO ENVIO ++++++"/>
        </log>
        <switch source="$ctx:flag_supera_fecha">
            <case regex="NO">
                <log description="Continuar">
                    <property name="LOG" value="+++++++ CONTINUAR CON REINTENTO ENVIO ++++++"/>
                </log>
            </case>
            <default>
                <log description="Actualizar estado reintento">
                    <property name="LOG" value="+++++++ ACTUALIZAR ESTADO REINTENTO ENVIO
+++++++++"/>
                </log>
                <dbreport description="Update reintento">
                    <connection>
                        <pool>
                            <driver>org.postgresql.Driver</driver>
                            <url>${SYSTEM:DB_PG_URL}</url>
                            <user>${SYSTEM:DB_PG_USR}</user>
                            <password>${SYSTEM:DB_PG_PSW}</password>
                        </pool>
                    </connection>
                    <statement>
                        <sql><![CDATA[INSERT INTO tbl_sda_reintento_envio (numero_transmision, codigo_transmision,
informacion_transmision, nro_reintento, cpceestadoenvio, cpdfechasgtereintento) VALUES (?, ?, ?, ?, ?,
TO_TIMESTAMP( ?, 'YYYY-MM-DD"T"HH24:MI:SS'))]]></sql>
                            <parameter expression="$ctx:PROP_NUMTRANSMISION" type="VARCHAR"/>
                            <parameter expression="$ctx:PROP_CODTRANSMISION" type="VARCHAR"/>
                            <parameter expression="$ctx:PROP_INFOTRANSMISION" type="VARCHAR"/>
                            <parameter expression="$ctx:PROP_NUMREINTENTO" type="VARCHAR"/>
                            <parameter type="VARCHAR" value="true"/>
                            <parameter expression="$ctx:PROP_FECHASGTEREINTENTO" type="VARCHAR"/>
                        </statement>
                    </dbreport>
                    <drop description="End flow"/>
                </default>
            </switch>
        </default>
    </switch>
</switch>
<switch source="$ctx:PROP_CODTRANSMISION">

```

```

<case regex="0121|0122">
  <log description="IRM">
    <property name="LOG" value="+++++++ IRM +++++++"/>
  </log>
  <dblookup description="Diferencia minutos">
    <connection>
      <pool>
        <driver>org.postgresql.Driver</driver>
        <url>${SYSTEM:DB_PG_URL}</url>
        <user>${SYSTEM:DB_PG_USR}</user>
        <password>${SYSTEM:DB_PG_PSW}</password>
      </pool>
    </connection>
    <statement>
      <sql><![CDATA[SELECT (DATE_PART('day', timezone('America/Lima', CURRENT_TIMESTAMP) -
cpdfechaoperacion) * 24 + DATE_PART('hour', timezone('America/Lima', CURRENT_TIMESTAMP) -
cpdfechaoperacion) * 60 + DATE_PART('minute', timezone('America/Lima', CURRENT_TIMESTAMP) -
cpdfechaoperacion) AS diff_min, (DATE_PART('day', timezone('America/Lima', CURRENT_TIMESTAMP) -
cpdfechaoperacion) * 24 + DATE_PART('hour', timezone('America/Lima', CURRENT_TIMESTAMP) -
cpdfechaoperacion) * 60 + DATE_PART('minute', timezone('America/Lima', CURRENT_TIMESTAMP) -
cpdfechaoperacion) < ? AS result_diff_min FROM tbl_sda_irm WHERE cpnid_irm = ?]]></sql>
      <parameter expression="${ctx:PROP_LIMITEENVIO}" type="INTEGER"/>
      <parameter expression="${ctx:PROP_NUMTRANSMISION}" type="INTEGER"/>
      <result column="diff_min" name="diff_min"/>
      <result column="result_diff_min" name="result_diff_min"/>
    </statement>
  </dblookup>
  <propertyGroup description="Minutos">
    <property expression="get-property('diff_min')" name="PROP_DIFFMIN" scope="default"
type="STRING"/>
    <property expression="get-property('result_diff_min')" name="PROP_RESULTDIFFMIN" scope="default"
type="STRING"/>
  </propertyGroup>
</case>
<case regex="RSA">
  <log description="RSA">
    <property name="LOG" value="+++++++ RSA +++++++"/>
  </log>
  <sequence key="ReintentoEnvioSunatRSASeq"/>
  <drop description="End flow"/>
</case>
<default>
  <drop description="End flow"/>
</default>
</switch>
<log description="Log minutos">
  <property name="LOG" value="+++++++ MINUTOS PASADOS +++++++"/>
  <property expression="${ctx:PROP_DIFFMIN}" name="MINUTOS"/>
  <property expression="${ctx:PROP_RESULTDIFFMIN}" name="CONTINUAR"/>
</log>
<switch source="${ctx:PROP_RESULTDIFFMIN}">
  <case regex="true">
    <log description="Continuar">
      <property name="LOG" value="+++++++ CONTINUAR CON REINTENTO ENVIO +++++++"/>
    </log>
    <sequence key="ReintentoEnvioSunatIRMSeq"/>
    <drop description="End flow"/>
  </case>
  <default>
    <log description="Detener">
      <property name="LOG" value="+++++++ DETENER REINTENTO ENVIO +++++++"/>
    </log>
    <dbreport description="Insert control">

```

```

<connection>
  <pool>
    <driver>org.postgresql.Driver</driver>
    <url>${SYSTEM:DB_PG_URL}</url>
    <user>${SYSTEM:DB_PG_USR}</user>
    <password>${SYSTEM:DB_PG_PSW}</password>
  </pool>
</connection>
<statement>
  <sql><![CDATA[INSERT INTO tbl_sda_control (cpnnumerotransmision, cpccodtransmision,
cpnid_evento, cpcusuario_servicio) VALUES (?, ?, ?, ?)]]></sql>
  <parameter expression="${Ctx:PROP_NUMTRANSMISION}" type="INTEGER"/>
  <parameter expression="${Ctx:PROP_CODTRANSMISION}" type="CHAR"/>
  <parameter type="INTEGER" value="8"/>
  <parameter type="VARCHAR" value="SERV_REINTENTO_TRANSMISION"/>
</statement>
</dbreport>
<log description="Email">
  <property name="LOG" value="+++++++ ENVIAR ALERTA ++++"/>
  <property expression="${Ctx:PROP_EMAIL_RECIPIENTS}" name="EMAIL(S)"/>
</log>
<property description="Email content" expression="fn:concat('Se excedio el limite de tiempo de ',
${Ctx:PROP_LIMITEENVIO}, ' minutos para realizar la transmision. ', 'Datos de transmision: ',
${Ctx:PROP_INFOTRANSMISION})" name="PROP_EMAIL_CONTENT" scope="default" type="STRING"/>
<class description="Enviar Alerta" name="pe.com.tpp.mediators.reintentotransmision.EnviarAlerta">
  <property name="subject" value="ALERTA! Falla de conexion y sin acuse de recepcion"/>
  <property expression="${Ctx:PROP_EMAIL_CONTENT}" name="content"/>
  <property expression="${Ctx:PROP_EMAIL_RECIPIENTS}" name="recipients"/>
  <property name="user" value="${Ctx:PROP_USR}"/>
  <property name="password" value="${Ctx:PROP_PSW}"/>
  <property name="host" value="${Ctx:PROP_HOST}"/>
  <property name="from" value="enviocorreo@tpp.com.pe"/>
</class>
<log description="End retry">
  <property name="LOG" value="+++++++ FIN DE REINTENTO ++++"/>
</log>
<drop description="End flow"/>
</default>
</switch>
</sequence>

```

Tabla 28. Código de flujo de proceso ReintentoEnvioMainErrorSeq

```

<sequence name="ReintentoEnvioMainErrorSeq" trace="disable" xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ REINTENTO ENVIO (MAIN ERROR SEQUENCE) ++++"/>
    <property expression="get-property('ERROR_CODE')" name="ERROR_CODE"/>
    <property expression="get-property('ERROR_MESSAGE')" name="ERROR_MESSAGE"/>
    <property expression="get-property('ERROR_DETAIL')" name="ERROR_DETAIL"/>
    <property expression="get-property('ERROR_EXCEPTION')" name="ERROR_EXCEPTION"/>
  </log>
  <drop description="End flow"/>
</sequence>

```

Tabla 29. Código de flujo de proceso ReintentoEnvioSunatIRMSeq

```

<?xml version="1.0" encoding="UTF-8"?>
<sequence name="ReintentoEnvioSunatIRMSeq" onError="ReintentoEnvioSunatIRMErrSeq" trace="disable"
xmlns="http://ws.apache.org/ns/synapse">

```

```

<log description="Log sequence">
  <property name="LOG" value="+++++++ REINTENTO ENVIO SUNAT IRM (SEQUENCE) +++++++"/>
  <property expression="$ctx:PROP_NUMTRANSMISION" name="NUMERO_TRANSMISION"/>
  <property expression="$ctx:PROP_CODTRANSMISION" name="CODIGO_TRANSMISION"/>
  <property expression="$ctx:PROP_INFOTRANSMISION" name="INFO_TRANSMISION"/>
  <property expression="$ctx:PROP_CODOMA" name="CODIGO_OMA"/>
</log>
<payloadFactory description="Create payload" media-type="xml">
  <format>
    <soapenv:Envelope xmlns:ser="http://services.sigad.sunat.gob.pe"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
      <soapenv:Body>
        <recibirArchivo xmlns="http://services.sigad.sunat.gob.pe">
          <numeroTransaccion xmlns="">$3</numeroTransaccion>
          <informacionArchivo xmlns="">$4</informacionArchivo>
        </recibirArchivo>
      </soapenv:Body>
    </soapenv:Envelope>
  </format>
  <args>
    <arg evaluator="xml" expression="$ctx:PROP_CODOMA"/>
    <arg evaluator="xml" expression="$ctx:PROP_ARCHIVOBASE64"/>
  </args>
</payloadFactory>
<property description="Petición envío" expression="/*" name="PROP_PETICIONENVIO" scope="default"
type="STRING"/>
<dbreport description="Insert reintento">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <uri>${SYSTEM:DB_PG_URL}</uri>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[INSERT INTO tbl_sda_reintento (cpnumerotransmision, cpccodtransmision,
cpdfechareintento, cpctiporeintento, cpcpeticionreintentoxml) VALUES (?, ?, CURRENT_TIMESTAMP, ?,
?)]]></sql>
    <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
    <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
    <parameter type="VARCHAR" value="SERV_REINTENTO_TRANSMISION"/>
    <parameter expression="$ctx:PROP_PETICIONENVIO" type="VARCHAR"/>
  </statement>
</dbreport>
<property description="Content type" name="ContentType" scope="axis2" type="STRING" value="text/xml"/>
<call>
  <endpoint>
    <wsdl port="ReceptorWebServiceServiceImplPort" service="ReceptorService.htm"
uri="${SYSTEM:WSDL_URI}">
      </wsdl>
    </endpoint>
  </call>
</log description="Log response payload" level="full"/>
<dbreport description="Update monitoreo">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <uri>${SYSTEM:DB_PG_URL}</uri>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>

```

```

<statement>
  <sql><![CDATA[UPDATE tbl_sda_monitoreo SET cpcacuseenvioxml = ?, cpcanhoenvio = ?,
cpdfecharecepcionenvio = TO_TIMESTAMP( ?, 'YYYY-MM-DD"HH24:MI:SS'), cpcticketenvio = ? WHERE
cpnnumerotransmision = ? AND cpccodtransmision = ?]]></sql>
  <parameter expression="/*" type="VARCHAR"/>
  <parameter expression="/*[local-name()='anhoEnvio']/text()" type="CHAR"/>
  <parameter expression="/*[local-name()='fechaRecepcion']/text()" type="VARCHAR"/>
  <parameter expression="/*[local-name()='ticketEnvio']/text()" type="VARCHAR"/>
  <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
  <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
</statement>
</dbreport>
<property description="Ticket envio" expression="/*[local-name()='ticketEnvio']/text()"
name="PROP_TICKETENVIO" scope="default" type="STRING"/>
<property description="Anho envio" expression="/*[local-name()='anhoEnvio']/text()"
name="PROP_ANHOENVIO" scope="default" type="STRING"/>
<payloadFactory description="Create kafka payload" media-type="json">
  <format>
    {"numero_transmision":"$1", "codigo_transmision":"$2", "ticket_envio":"$3",
"anho_envio":"$4"}</format>
  <args>
    <arg evaluator="xml" expression="$ctx:PROP_NUMTRANSMISION"/>
    <arg evaluator="xml" expression="$ctx:PROP_CODTRANSMISION"/>
    <arg evaluator="xml" expression="$ctx:PROP_TICKETENVIO"/>
    <arg evaluator="xml" expression="$ctx:PROP_ANHOENVIO"/>
  </args>
</payloadFactory>
<log description="Log json payload" level="full"/>
<property description="Content type" name="ContentType" scope="axis2" type="STRING"
value="application/json"/>
<script description="Environment variables"
language="js"><![CDATA[mc.setProperty("PROP_KAFKA_BOOTSTRAP_SERVERS",
java.lang.System.getenv("KAFKA_BOOTSTRAP_SERVERS"));]]></script>
  <kafkaTransport.init>
    <bootstrapServers>{$ctx:PROP_KAFKA_BOOTSTRAP_SERVERS}</bootstrapServers>
  </kafkaTransport.init>
  <kafkaTransport.publishMessages>
    <topic>tpp_transmision_consulta</topic>
  </kafkaTransport.publishMessages>
  <drop description="End flow"/>
</sequence>

```

Tabla 30. Código de flujo de proceso ReintentoEnvioSunatIRMErrSeq

```

<sequence name="ReintentoEnvioSunatIRMErrSeq" trace="disable"
xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ REINTENTO ENVIO SUNAT IRM (ERROR SEQUENCE)
+++++++"/>
    <property expression="get-property('ERROR_CODE')" name="ERROR_CODE"/>
    <property expression="get-property('ERROR_MESSAGE')" name="ERROR_MESSAGE"/>
    <property expression="get-property('ERROR_DETAIL')" name="ERROR_DETAIL"/>
    <property expression="get-property('ERROR_EXCEPTION')" name="ERROR_EXCEPTION"/>
  </log>
  <log description="Send retry">
    <property name="LOG" value="+++++++ REINTENTAR ENVIO SUNAT IRM ++++++++"/>
  </log>
  <script description="Environment variables" language="js"><![CDATA[mc.setProperty("PROP_DB_PG_URL",
java.lang.System.getenv("DB_PG_URL"));mc.setProperty("PROP_DB_PG_USERNAME",

```

```

java.lang.System.getenv("DB_PG_USERNAME"));mc.setProperty("PROP_DB_PG_PASSWORD",
java.lang.System.getenv("DB_PG_PASSWORD"));]]></script>
<class description="Validar Reintento" name="pe.com.tpp.mediators.reintentotransmision.ValidarReintento">
  <property name="db_driver" value="org.postgresql.Driver"/>
  <property expression="$ctx:PROP_DB_PG_URL" name="db_url"/>
  <property expression="$ctx:PROP_DB_PG_USERNAME" name="db_username"/>
  <property expression="$ctx:PROP_DB_PG_PASSWORD" name="db_password"/>
  <property expression="$ctx:PROP_CODTRANSMISION" name="cpccod_transmision"/>
  <property expression="$ctx:PROP_NUMTRANSMISION" name="cpnid_numerotransmision"/>
  <property expression="$ctx:PROP_NUMREINTENTO" name="nro_reintento"/>
</class>
<switch source="$ctx:flag_supera_limite">
  <case regex="NO">
    <log description="No supera">
      <property name="LOG" value="+++++++ NO SUPERA EL LIMITE DE REINTENTO +++++++"/>
      <property expression="$ctx:flag_supera_limite" name="FLAG-SUPERA-LIMITE"/>
    </log>
  </case>
  <default>
    <log description="Default">
      <property name="LOG" value="+++++++ SUPERA EL LIMITE DE REINTENTO > 3 +++++++"/>
      <property expression="$ctx:flag_supera_limite" name="FLAG-SUPERA-LIMITE"/>
    </log>
    <log description="Reintento">
      <property name="LOG" value="+++++++ GUARDAR NUEVO REINTENTO +++++++"/>
    </log>
    <dbreport description="Insert Reintento">
      <connection>
        <pool>
          <driver>org.postgresql.Driver</driver>
          <url>${SYSTEM:DB_PG_URL}</url>
          <user>${SYSTEM:DB_PG_USR}</user>
          <password>${SYSTEM:DB_PG_PSW}</password>
        </pool>
      </connection>
      <statement>
        <sql><![CDATA[INSERT INTO tbl_sda_reintento_envio (numero_transmision, codigo_transmision,
informacion_transmision, nro_reintento, cpcestadoenvio, cpdfechasgtereintento) VALUES (?, ?, ?, ?, ?,
TO_TIMESTAMP( ?, 'YYYY-MM-DD"HH24:MI:SS'))]]></sql>
        <parameter expression="$ctx:PROP_NUMTRANSMISION" type="VARCHAR"/>
        <parameter expression="$ctx:PROP_CODTRANSMISION" type="VARCHAR"/>
        <parameter expression="$ctx:PROP_INFOTRANSMISION" type="VARCHAR"/>
        <parameter expression="$ctx:sgte_nro_reintento" type="VARCHAR"/>
        <parameter type="VARCHAR" value="true"/>
        <parameter expression="$ctx:fecha_sgte_reintento" type="VARCHAR"/>
      </statement>
    </dbreport>
    <drop description="End flow"/>
  </default>
</switch>
<payloadFactory description="Create kafka payload" media-type="json">
  <format>
    {"numero_transmision": "$1", "codigo_transmision": "$2", "informacion_transmision": "$3",
"nro_reintento": "$4"}</format>
    <args>
      <arg evaluator="xml" expression="$ctx:PROP_NUMTRANSMISION"/>
      <arg evaluator="xml" expression="$ctx:PROP_CODTRANSMISION"/>
      <arg evaluator="xml" expression="$ctx:PROP_INFOTRANSMISION"/>
      <arg evaluator="xml" expression="$ctx:sgte_nro_reintento"/>
    </args>
  </payloadFactory>
  <log description="Log json payload" level="full"/>

```

```

<property description="Content type" name="ContentType" scope="axis2" type="STRING"
value="application/json"/>
<script description="Environment variables"
language="js"><![CDATA[mc.setProperty("PROP_KAFKA_BOOTSTRAP_SERVERS",
java.lang.System.getenv("KAFKA_BOOTSTRAP_SERVERS"));]]></script>
<kafkaTransport.init>
  <bootstrapServers>${ctx:PROP_KAFKA_BOOTSTRAP_SERVERS}</bootstrapServers>
</kafkaTransport.init>
<kafkaTransport.publishMessages>
  <topic>tpp_reintento_envio</topic>
</kafkaTransport.publishMessages>
<drop description="End transmision"/>
</sequence>

```

Tabla 31. Código de flujo de proceso ReintentoEnvioSunatRSASeq

```

<sequence name="ReintentoEnvioSunatRSASeq" onError="ReintentoEnvioSunatRSAErrorSeq" trace="disable"
xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ REINTENTO ENVIO SUNAT RSA (SEQUENCE) +++++++"/>
    <property expression="${ctx:PROP_NUMTRANSMISION}" name="NUMERO_TRANSMISION"/>
    <property expression="${ctx:PROP_CODTRANSMISION}" name="CODIGO_TRANSMISION"/>
    <property expression="${ctx:PROP_INFOTRANSMISION}" name="INFO_TRANSMISION"/>
  </log>
  <dblookup description="Diferencia minutos">
    <connection>
      <pool>
        <driver>org.postgresql.Driver</driver>
        <url>${SYSTEM:DB_PG_URL}</url>
        <user>${SYSTEM:DB_PG_USR}</user>
        <password>${SYSTEM:DB_PG_PSW}</password>
      </pool>
    </connection>
    <statement>
      <sql><![CDATA[SELECT (DATE_PART('day', timezone('America/Lima', CURRENT_TIMESTAMP) -
cpdfechaoperacion) * 24 + DATE_PART('hour', timezone('America/Lima', CURRENT_TIMESTAMP) -
cpdfechaoperacion)) * 60 + DATE_PART('minute', timezone('America/Lima', CURRENT_TIMESTAMP) -
cpdfechaoperacion) AS diff_min, (DATE_PART('day', timezone('America/Lima', CURRENT_TIMESTAMP) -
cpdfechaoperacion) * 24 + DATE_PART('hour', timezone('America/Lima', CURRENT_TIMESTAMP) -
cpdfechaoperacion)) * 60 + DATE_PART('minute', timezone('America/Lima', CURRENT_TIMESTAMP) -
cpdfechaoperacion) < ? AS result_diff_min FROM tbl_sda_irm WHERE cpnid_irm = ?]]></sql>
      <parameter expression="${ctx:PROP_LIMITEENVIO}" type="INTEGER"/>
      <parameter expression="${ctx:PROP_NUMTRANSMISION}" type="INTEGER"/>
      <result column="diff_min" name="diff_min"/>
      <result column="result_diff_min" name="result_diff_min"/>
    </statement>
  </dblookup>
  <property description="Minutes" expression="get-property('diff_min')" name="PROP_DIFFMIN"
scope="default" type="STRING"/>
  <property description="Result minutes" expression="get-property('result_diff_min')"
name="PROP_RESULTDIFFMIN" scope="default" type="STRING"/>
  <log description="Log minutes">
    <property name="LOG" value="+++++++ MINUTOS PASADOS +++++++"/>
    <property expression="${ctx:PROP_DIFFMIN}" name="MINUTOS"/>
    <property expression="${ctx:PROP_RESULTDIFFMIN}" name="CONTINUAR"/>
  </log>
  <dblookup description="Select Archivo XML">
    <connection>
      <pool>
        <driver>org.postgresql.Driver</driver>
        <url>${SYSTEM:DB_PG_URL}</url>

```

```

    <user>$$SYSTEM:DB_PG_USR</user>
    <password>$$SYSTEM:DB_PG_PSW</password>
  </pool>
</connection>
<statement>
  <sql><![CDATA[SELECT cpcarchivoenvioxml FROM tbl_sda_monitoreo WHERE cpnumerotransmision = ?
AND cpcodtransmision = ?]]></sql>
  <parameter expression="$$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
  <parameter expression="$$ctx:PROP_CODTRANSMISION" type="CHAR"/>
  <result column="cpcarchivoenvioxml" name="cpcarchivoenvioxml"/>
</statement>
</dblookup>
<property description="Archivo XML" expression="get-property('cpcarchivoenvioxml')"
name="PROP_ARCHIVOENVIOMUX" scope="default" type="STRING"/>
<payloadFactory description="Create payload" media-type="xml">
  <format>
    <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:wco.datamodel:pe:gob:sunat:controladuanero:ingreso:reconfisico:DeclaracionReconocimientoFi
sicoRequest:1" xmlns:urn1="urn:wco.datamodel:WCO:Declaration_DS:1">
      <soapenv:Body>
        $3
      </soapenv:Body>
    </soapenv:Envelope>
  </format>
  <args>
    <arg evaluator="xml" expression="$$ctx:PROP_ARCHIVOENVIOMUX"/>
  </args>
</payloadFactory>
<property description="Petición envío" expression="/*" name="PROP_PETICIONENVIO" scope="default"
type="STRING"/>
<dbreport description="Insert reintento">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <url>$$SYSTEM:DB_PG_URL</url>
      <user>$$SYSTEM:DB_PG_USR</user>
      <password>$$SYSTEM:DB_PG_PSW</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[INSERT INTO tbl_sda_reintento (cpnumerotransmision, cpcodtransmision,
cpdfechareintento, cpctiporeintento, cpcpeticionreintentoxml) VALUES (?, ?, CURRENT_TIMESTAMP, ?,
?)]></sql>
    <parameter expression="$$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
    <parameter expression="$$ctx:PROP_CODTRANSMISION" type="CHAR"/>
    <parameter type="VARCHAR" value="SERV_REINTENTO_TRANSMISION"/>
    <parameter expression="$$ctx:PROP_PETICIONENVIO" type="VARCHAR"/>
  </statement>
</dbreport>
<property description="Content type" name="ContentType" scope="axis2" type="STRING" value="text/xml"/>
<call>
  <endpoint>
    <wsdl port="ReconocimientoFisicoPort" service="ReconocimientoFisicoService"
uri="https://ws.sunat.gob.pe/ws/controladuanero/ReconocimientoFisicoService.htm?wsdl">
      </wsdl>
    </endpoint>
  </call>
  <log description="Log response payload" level="full"/>
  <property description="Respuesta RSA" expression="/*" name="PROP_RESPUESTARSA" scope="default"
type="STRING"/>
  <class description="Transform response"
name="pe.com.tpp.mediators.reintentotransmision.TransformResponse">
    <property expression="/*" name="rsaResponse"/>

```



```

</class>
<log description="Log response RSA" level="custom">
  <property expression="$ctx:statusCode" name="statusCode"/>
  <property expression="$ctx:informacionMensaje" name="informacionMensaje"/>
  <property expression="$ctx:responseXML" name="responseXML"/>
</log>
<switch source="$ctx:statusCode">
  <case regex="1">
    <log description="Case 1">
      <property name="LOG" value="+++++++ STATUS-CODE: 1 +++++++"/>
    </log>
    <property description="Evento 13" name="PROP_EVENTOID" scope="default" type="STRING"
value="13"/>
  </case>
  <default>
    <log description="Default">
      <property name="LOG" value="+++++++ STATUS-CODE:8 +++++++"/>
      <property expression="$ctx:statusCode" name="STATUS-CODE"/>
    </log>
    <property description="Evento 9" name="PROP_EVENTOID" scope="default" type="STRING" value="9"/>
  </default>
</switch>
<log description="Send response to on-premise">
  <property name="LOG" value="+++++++ ENVIAR RESPUESTA DE RSA +++++++"/>
</log>
<dblookup description="Select data transmision">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <url>${SYSTEM:DB_PG_URL}</url>
      <user>${SYSTEM:DB_PG_USR}</user>
      <password>${SYSTEM:DB_PG_PSW}</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[SELECT cpnid_tticket,cpnid_envioaduana FROM tbl_sda_rsa WHERE cpnid_rsa = ? AND
cpccodtransmision = ?]]></sql>
    <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
    <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
    <result column="cpnid_tticket" name="cpnid_tticket"/>
    <result column="cpnid_envioaduana" name="cpnid_envioaduana"/>
  </statement>
</dblookup>
<payloadFactory description="Enviar respuesta" media-type="json">
  <format>
    {"numero_transmision": "$1", "codigo_transmision": "$2", "codigo_respuesta": "$3",
"mensaje_respuesta": "$4", "id_ticket": "$5", "id_envioaduana": "$6", "respuesta_xml": "$7"}</format>
  <args>
    <arg evaluator="xml" expression="$ctx:PROP_NUMTRANSMISION"/>
    <arg evaluator="xml" expression="$ctx:PROP_CODTRANSMISION"/>
    <arg evaluator="xml" expression="$ctx:statusCode"/>
    <arg evaluator="xml" expression="$ctx:informacionMensaje"/>
    <arg evaluator="xml" expression="$ctx:cpnid_tticket"/>
    <arg evaluator="xml" expression="$ctx:cpnid_envioaduana"/>
    <arg evaluator="xml" expression="$ctx:responseXMLBase64"/>
  </args>
</payloadFactory>
<log description="Log payload" level="full"/>
<property description="Content type" name="ContentType" scope="axis2" type="STRING"
value="application/json"/>
<script description="Environment variables"
language="js"><![CDATA[mc.setProperty("PROP_KAFKA_BOOTSTRAP_SERVERS",
java.lang.System.getenv("KAFKA_BOOTSTRAP_SERVERS"));]]></script>

```

```

<kafkaTransport.init>
  <bootstrapServers>${ctx:PROP_KAFKA_BOOTSTRAP_SERVERS}</bootstrapServers>
</kafkaTransport.init>
<kafkaTransport.publishMessages>
  <topic>tpp_transmission_resultado</topic>
</kafkaTransport.publishMessages>
<log description="Log end">
  <property name="LOG" value="+++++++ FIN DE TRANSMISION ++++++"/>
</log>
<drop description="End transmision"/>
</sequence>

```

Tabla 32. Código de flujo de proceso ReintentoEnvioSunatRSAErrorSeq

```

<sequence name="ReintentoEnvioSunatRSAErrorSeq" trace="disable"
xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ REINTENTO ENVIO SUNAT RSA (ERROR SEQUENCE) ++++++"/>
    <property expression="get-property('ERROR_CODE')" name="ERROR_CODE"/>
    <property expression="get-property('ERROR_MESSAGE')" name="ERROR_MESSAGE"/>
    <property expression="get-property('ERROR_DETAIL')" name="ERROR_DETAIL"/>
    <property expression="get-property('ERROR_EXCEPTION')" name="ERROR_EXCEPTION"/>
  </log>
  <log description="Send retry">
    <property name="LOG" value="+++++++ REINTENTAR ENVIO SUNAT RSA ++++++"/>
  </log>
  <script description="Environment variables" language="js"><![CDATA[mc.setProperty("PROP_DB_PG_URL",
java.lang.System.getenv("DB_PG_URL"));mc.setProperty("PROP_DB_PG_USERNAME",
java.lang.System.getenv("DB_PG_USERNAME"));mc.setProperty("PROP_DB_PG_PASSWORD",
java.lang.System.getenv("DB_PG_PASSWORD"));]]></script>
  <class description="Validador Reintento" name="pe.com.tpp.mediators.reintentotransmision.ValidadorReintento">
    <property name="db_driver" value="org.postgresql.Driver"/>
    <property expression="${ctx:PROP_DB_PG_URL}" name="db_url"/>
    <property expression="${ctx:PROP_DB_PG_USERNAME}" name="db_username"/>
    <property expression="${ctx:PROP_DB_PG_PASSWORD}" name="db_password"/>
    <property expression="${ctx:PROP_CODTRANSMISION}" name="cpccod_transmision"/>
    <property expression="${ctx:PROP_NUMTRANSMISION}" name="cpnid_numero transmision"/>
    <property expression="${ctx:PROP_NUMREINTENTO}" name="nro_reintento"/>
  </class>
  <switch source="${ctx:flag_supera_limite">
    <case regex="NO">
      <log description="No supera">
        <property name="LOG" value="+++++++ NO SUPERA EL LIMITE DE REINTENTO ++++++"/>
        <property expression="${ctx:flag_supera_limite}" name="FLAG-SUPERA-LIMITE"/>
      </log>
    </case>
    <default>
      <log description="Default">
        <property name="LOG" value="+++++++ SUPERA EL LIMITE DE REINTENTO > 3 ++++++"/>
        <property expression="${ctx:flag_supera_limite}" name="FLAG-SUPERA-LIMITE"/>
      </log>
      <log description="Reintento">
        <property name="LOG" value="+++++++ GUARDAR NUEVO REINTENTO ++++++"/>
      </log>
      <dbreport description="Insert Reintento">
        <connection>
          <pool>
            <driver>org.postgresql.Driver</driver>
            <url>${SYSTEM:DB_PG_URL}</url>
            <user>${SYSTEM:DB_PG_USR}</user>
            <password>${SYSTEM:DB_PG_PSW}</password>
          </pool>
        </connection>
      </dbreport>
    </default>
  </switch>

```

```

    </pool>
  </connection>
  <statement>
    <sql><![CDATA[INSERT INTO tbl_sda_reintento_envio (numero_transmision, codigo_transmision,
informacion_transmision, nro_reintento, cpceestadoenvio, cpdfechasgtereintento) VALUES (?, ?, ?, ?, ?,
TO_TIMESTAMP( ?, 'YYYY-MM-DD"HH24:MI:SS'))]]></sql>
    <parameter expression="$ctx:PROP_NUMTRANSMISION" type="VARCHAR"/>
    <parameter expression="$ctx:PROP_CODTRANSMISION" type="VARCHAR"/>
    <parameter expression="$ctx:PROP_INFOTRANSMISION" type="VARCHAR"/>
    <parameter expression="$ctx:sgte_nro_reintento" type="VARCHAR"/>
    <parameter type="VARCHAR" value="true"/>
    <parameter expression="$ctx:fecha_sgte_reintento" type="VARCHAR"/>
  </statement>
</dbreport>
<drop description="End flow"/>
</default>
</switch>
<payloadFactory description="Create kafka payload" media-type="json">
  <format>
    {"numero_transmision": "$1", "codigo_transmision": "$2", "informacion_transmision": "$3",
"nro_reintento": "$4"}</format>
    <args>
      <arg evaluator="xml" expression="$ctx:PROP_NUMTRANSMISION"/>
      <arg evaluator="xml" expression="$ctx:PROP_CODTRANSMISION"/>
      <arg evaluator="xml" expression="$ctx:PROP_INFOTRANSMISION"/>
      <arg evaluator="xml" expression="$ctx:sgte_nro_reintento"/>
    </args>
  </payloadFactory>
  <log description="Log json payload" level="full"/>
  <property description="ContentType" name="ContentType" scope="axis2" type="STRING"
value="application/json"/>
  <script description="Environment variables"
language="js"><![CDATA[mc.setProperty("PROP_KAFKA_BOOTSTRAP_SERVERS",
java.lang.System.getenv("KAFKA_BOOTSTRAP_SERVERS"));]]></script>
  <kafkaTransport.init>
    <bootstrapServers>{$ctx:PROP_KAFKA_BOOTSTRAP_SERVERS}</bootstrapServers>
  </kafkaTransport.init>
  <kafkaTransport.publishMessages>
    <topic>tpp_reintento_envio</topic>
  </kafkaTransport.publishMessages>
  <drop description="End transmision"/>
</sequence>

```

-Desarrollo de Componente Reintento Consulta

-Entrada de evento (Inbound-Endpoints)

Se crearán las siguientes entradas de evento:

- "ReintentoConsultaInboundEP.xml" que leerá las entradas (Formato JSON) enviados a la cola de mensajes "tpp_reintento_consulta". Por cada lectura exitosa de documento seguirá el flujo de proceso "ReintentoConsultaMainSeq" (Sequence). En caso de lectura errónea seguirá el flujo de proceso "ReintentoConsultaMainErrorSeq" (Sequence).

Se define los parámetros de configuración para la entrada de evento "ReintentoConsultaInboundEP.xml".

- Sequential: Lectura secuencial de los mensajes de entrada. Valor configurado: true
- Interval: Intervalo de lectura de los mensajes de entrada en milisegundos. Valor configurado: 5000
- Coordination: Ejecución en modo cluster (false) o en un solo nodo (true). Valor configurado: true
- Inbound.behavior: Comportamiento de lectura de datos. Valor configurado: pooling
- Topic.name: Nombre de la cola de mensajes de entrada para la lectura de transmisiones. Valor configurado: tpp_reintento_consulta
- Poll.timeout: Tiempo máximo de espera de lectura en milisegundos. Valor configurado: 1000
- Max.poll.records: Cantidad máxima de mensajes a lectura de forma periódica. Valor configurado: 5
- ContentType: Formato de mensajes de entrada. Valor configurado: application/json

Tabla 33. Código de entrada de evento ReintentoConsultaInboundEP

```
<inboundEndpoint class="org.wso2.carbon.inbound.kafka.KafkaMessageConsumer"
name="ReintentoConsultaInboundEP" onError="ReintentoConsultaMainErrorSeq"
sequence="ReintentoConsultaMainSeq" suspend="false" xmlns="http://ws.apache.org/ns/synapse">
  <parameters>
    <parameter name="sequential">true</parameter>
    <parameter name="interval">5000</parameter>
    <parameter name="coordination">true</parameter>
    <parameter name="inbound.behavior">polling</parameter>
    <parameter name="topic.name">tpp_reintento_consulta</parameter>
    <parameter name="poll.timeout">1000</parameter>
    <parameter name="max.poll.records">5</parameter>
    <parameter name="bootstrap.servers">${SYSTEM:KAFKA_BOOTSTRAP_SERVERS}</parameter>
    <parameter name="group.id">reintentoconsultawso2service</parameter>
    <parameter name="contentType">application/json</parameter>
  </parameters>
</inboundEndpoint>
```

- “ReintentoConsultaDBEventInboundEP.xml” que leerá las entradas (Formato JSON) desde la tabla de reintendo de consulta pendientes “tbl_sda_reintento_consulta”. Por cada lectura exitosa de documento seguirá el flujo de proceso “ReintentoConsultaMainSeq” (Sequence). En caso de lectura errónea seguirá el flujo de proceso “ReintentoConsultaMainErrorSeq” (Sequence).

Se define los parámetros de configuración para la entrada de evento “ReintentoEnvioDBEventInboundEP.xml”.

- Sequential: Lectura secuencial de los registros de las tablas. Valor configurado: true
- Interval: Intervalo de lectura de los mensajes de entrada en milisegundos. Valor configurado: 300000
- Coordination: Ejecución en modo cluster (false) o en un solo nodo (true). Valor configurado: true
- Inbound.behavior: Comportamiento de lectura de datos. Valor configurado: pooling

- driverName: Nombre del driver de conexión a la Base de Datos. Valor configurado: org.postgresql.Driver
- url: Cadena de conexión a Base de Datos. Valor configurado: \$SYSTEM:DB_PG_URL
- username: Usuario de conexión a Base de Datos. Valor configurado: \$SYSTEM:DB_PG_USERNAME
- password: Password de conexión a Base de Datos. Valor configurado: \$SYSTEM:DB_PG_PASSWORD
- tableName: Tabla de consulta de transmisiones pendientes de reintentar. Valor configurado: tbl_sda_reintento_consulta
- filteringCriteria: Criterio de filtro de registros. Valor configurado: byBooleanColumn
- cpceestadoenvio: Columna que aplica el criterio de filtro de registros. Valor configurado: cpceestadoenvio
- primaryKey: Llave primaria de la tabla. Valor configurado: cpnid_reintentoconsulta

Tabla 34. Código de entrada de evento ReintentoConsultaDBEventInboundEP

```

<inboundEndpoint class="org.wso2.carbon.inbound.poll.dbeventlistener.DBEventPollingConsumer"
name="ReintentoConsultaDBEventInboundEP" onError="ReintentoConsultaMainErrorSeq"
sequence="ReintentoConsultaMainSeq" suspend="false" xmlns="http://ws.apache.org/ns/synapse">
  <parameters>
    <parameter name="interval">300000</parameter>
    <parameter
name="class">org.wso2.carbon.inbound.poll.dbeventlistener.DBEventPollingConsumer</parameter>
    <parameter name="sequential">true</parameter>
    <parameter name="coordination">true</parameter>
    <parameter name="inbound.behavior">polling</parameter>
    <parameter name="driverName">org.postgresql.Driver</parameter>
    <parameter name="url">$SYSTEM:DB_PG_URL</parameter>
    <parameter name="username">$SYSTEM:DB_PG_USERNAME</parameter>
    <parameter name="password">$SYSTEM:DB_PG_PASSWORD</parameter>
    <parameter name="tableName">tbl_sda_reintento_consulta</parameter>
    <parameter name="filteringCriteria">byBooleanColumn</parameter>
    <parameter name="filteringColumnName">cpceestadoenvio</parameter>
    <parameter name="primaryKey">cpnid_reintentoconsulta</parameter>
    <parameter name="connectionValidationQuery">SELECT 1</parameter>
  </parameters>
</inboundEndpoint>

```

- Flujos de proceso (Sequence)

Se desarrollan dos flujos de procesos principales y 2 flujos secundarios:

ReintentoConsultaMainSeq: Contiene la lógica principal del flujo de proceso Consulta:

- Validación del límite de tiempo de consulta según configuración.
- Registros en tablas de eventos y control del proceso.
- Continuidad al subproceso "ReintentoConsultaSunatSeq".

ReintentoConsultaMainErrorSeq: Contiene la lógica de control de errores del flujo de proceso "ReintentoConsultaMainSeq":

- Registro de logs con el código y mensaje de error.

- Ejecuta el fin del flujo de proceso "ReintentoConsultaMainSeq".

ReintentoConsultaSunatSeq: Contiene la lógica principal del flujo de proceso:

- Creación de cliente de comunicación para reintentar la consulta de estado de transmisión al Web Services de SUNAT.
- Lectura de respuesta (Código y mensaje de respuesta) de Web Services de SUNAT.
- Registros en tablas de eventos y control del proceso.

ReintentoConsultaSunatErrorSeq: Contiene la lógica de control de errores del flujo de proceso "ReintentoConsultaSunatSeq":

- Registro de logs con el código y mensaje de error.
- Ejecuta el fin del flujo de proceso "ReintentoConsultaSunatSeq".

Tabla 35. Código de flujo de proceso ReintentoConsultaMainSeq

```
<sequence name="ReintentoConsultaMainSeq" onError="ReintentoConsultaMainErrorSeq" trace="disable"
xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log properties">
    <property name="LOG" value="+++++++ REINTENTO CONSULTA (MAIN SEQUENCE) +++++++"/>
    <property expression="json-eval($)" name="MENSAJE_JSON"/>
  </log>
  <propertyGroup description="Properties transmission">
    <property expression="//numero_transmission" name="PROP_NUMTRANSMISION" scope="default"
type="STRING"/>
    <property expression="//codigo_transmission" name="PROP_CODTRANSMISION" scope="default"
type="STRING"/>
    <property expression="//ticket_envio" name="PROP_TICKETENVIO" scope="default" type="STRING"/>
    <property expression="//anho_envio" name="PROP_ANHOENVIO" scope="default" type="STRING"/>
    <property expression="//nro_reintento" name="PROP_NUMREINTENTO" scope="default" type="STRING"/>
    <property expression="//cpdfechasgtereintento" name="PROP_FECHASGTEREINTENTO" scope="default"
type="STRING"/>
  </propertyGroup>
  <script description="Environment variables" language="js"><![CDATA[mc.setProperty("PROP_DB_PG_URL",
java.lang.System.getenv("DB_PG_URL"));mc.setProperty("PROP_DB_PG_USERNAME",
java.lang.System.getenv("DB_PG_USERNAME"));mc.setProperty("PROP_DB_PG_PASSWORD",
java.lang.System.getenv("DB_PG_PASSWORD"));]]></script>
  <class description="Validar Reintento fecha"
name="pe.com.tpp.mediators.reintentoconsulta.ValidarReintentoFecha">
    <property expression="$ctx:PROP_NUMREINTENTO" name="nro_reintento"/>
    <property expression="$ctx:PROP_FECHASGTEREINTENTO" name="fecha_sgte_reintento"/>
  </class>
  <switch source="$ctx:flag_reintento_cola">
    <case regex="SI">
      <log description="Continuar">
        <property name="LOG" value="+++++++ CONTINUAR CON REINTENTO CONSULTA +++++++"/>
      </log>
    </case>
    <default>
      <log description="Validar fecha de reintento">
        <property name="LOG" value="+++++++ EVALUAR FECHA REINTENTO CONSULTA +++++++"/>
      </log>
      <switch source="$ctx:flag_supera_fecha">
        <case regex="NO">
          <log description="Continuar">

```

```

        <property name="LOG" value="+++++++ CONTINUAR CON REINTENTO CONSULTA
+++++++"/>
    </log>
</case>
<default>
    <log description="Actualizar estado reintento">
        <property name="LOG" value="+++++++ ACTUALIZAR ESTADO REINTENTO CONSULTA
+++++++"/>
    </log>
    <dbreport description="Update reintento">
        <connection>
            <pool>
                <driver>org.postgresql.Driver</driver>
                <url>${SYSTEM:DB_PG_URL}</url>
                <user>${SYSTEM:DB_PG_USR}</user>
                <password>${SYSTEM:DB_PG_PSW}</password>
            </pool>
        </connection>
        <statement>
            <sql><![CDATA[INSERT INTO tbl_sda_reintento_consulta (numero_transmision,
codigo_transmision, ticket_envio, anho_envio, nro_reintento, cpcestadoenvio, cpdfechasgtereintento) VALUES
(?, ?, ?, ?, ?, TO_TIMESTAMP( ?, 'YYYY-MM-DD"T"HH24:MI:SS'))]></sql>
            <parameter expression="${ctx:PROP_NUMTRANSMISION}" type="VARCHAR"/>
            <parameter expression="${ctx:PROP_CODTRANSMISION}" type="VARCHAR"/>
            <parameter expression="${ctx:PROP_TICKETENVIO}" type="VARCHAR"/>
            <parameter expression="${ctx:PROP_ANHOENVIO}" type="VARCHAR"/>
            <parameter expression="${ctx:PROP_NUMREINTENTO}" type="VARCHAR"/>
            <parameter type="VARCHAR" value="true"/>
            <parameter expression="${ctx:PROP_FECHASGTEREINTENTO}" type="VARCHAR"/>
        </statement>
    </dbreport>
    <drop description="End flow"/>
</default>
</switch>
</default>
</switch>
<dblookup description="Select config">
    <connection>
        <pool>
            <driver>org.postgresql.Driver</driver>
            <url>${SYSTEM:DB_PG_URL}</url>
            <user>${SYSTEM:DB_PG_USR}</user>
            <password>${SYSTEM:DB_PG_PSW}</password>
        </pool>
    </connection>
    <statement>
        <sql><![CDATA[SELECT cpnlimreintentoconsulta, cpcsuariows, cpclavews FROM tbl_sda_configuracion
WHERE cpccodtransmision = ?]]></sql>
        <parameter expression="${ctx:PROP_CODTRANSMISION}" type="CHAR"/>
        <result column="cpclavews" name="cpclavews"/>
        <result column="cpnlimreintentoconsulta" name="cpnlimreintentoconsulta"/>
        <result column="cpcsuariows" name="cpcsuariows"/>
    </statement>
</dblookup>
<propertyGroup description="Properties config">
    <property expression="get-property('cpnlimreintentoconsulta')" name="PROP_LIMITECONSULTA"
scope="default" type="STRING"/>
    <property expression="get-property('cpcsuariows')" name="PROP_USUARIOWS" scope="default"
type="STRING"/>
    <property expression="get-property('cpclavews')" name="PROP_CLAVEWS" scope="default"
type="STRING"/>
</propertyGroup>
<log description="Evaluar reintento">

```

```

    <property name="LOG" value="+++++++ EVALUAR REINTENTO CONSULTA +++++++"/>
  </log>
  <dblookup description="Select monitoreo">
    <connection>
      <pool>
        <driver>org.postgresql.Driver</driver>
        <url>${SYSTEM:DB_PG_URL}</url>
        <user>${SYSTEM:DB_PG_USR}</user>
        <password>${SYSTEM:DB_PG_PSW}</password>
      </pool>
    </connection>
    <statement>
      <sql><![CDATA[SELECT (cpndeshabilitarreintento <> 0) OR (cpnidultimoevento = 14) AS
deshabilitarreintento, cpndeshabilitarreintento, COALESCE(cpnid_envioaduana::varchar, '') AS
cpnid_envioaduana FROM tbl_sda_monitoreo WHERE cpnnumerotransmision = ? AND cpccodtransmision =
?]]></sql>
      <parameter expression="${ctx:PROP_NUMTRANSMISION}" type="INTEGER"/>
      <parameter expression="${ctx:PROP_CODTRANSMISION}" type="CHAR"/>
      <result column="deshabilitarreintento" name="deshabilitarreintento"/>
      <result column="cpndeshabilitarreintento" name="cpndeshabilitarreintento"/>
      <result column="cpnid_envioaduana" name="cpnid_envioaduana"/>
    </statement>
  </dblookup>
  <propertyGroup description="Transmission data">
    <property expression="${ctx:cpnid_envioaduana}" name="PROP_IDENVIOADUANA" scope="default"
type="STRING"/>
    <property name="PROP_IDTICKET" scope="default" type="STRING" value=""/>
  </propertyGroup>
  <switch source="${ctx:deshabilitarreintento}">
    <case regex="false">
      <log description="Continuar">
        <property name="LOG" value="+++++++ CONTINUAR CON REINTENTO CONSULTA +++++++"/>
      </log>
    </case>
    <default>
      <log description="Detener">
        <property name="LOG" value="+++++++ DETENER REINTENTO CONSULTA +++++++"/>
      </log>
      <switch source="${ctx:cpndeshabilitarreintento}">
        <case regex="1">
          <dbreport description="Insert control">
            <connection>
              <pool>
                <driver>org.postgresql.Driver</driver>
                <url>${SYSTEM:DB_PG_URL}</url>
                <user>${SYSTEM:DB_PG_USR}</user>
                <password>${SYSTEM:DB_PG_PSW}</password>
              </pool>
            </connection>
            <statement>
              <sql><![CDATA[INSERT INTO tbl_sda_control (cpnnumerotransmision, cpccodtransmision,
cpnid_evento, cpcusuario_servicio) VALUES (?, ?, ?, ?)]]></sql>
              <parameter expression="${ctx:PROP_NUMTRANSMISION}" type="INTEGER"/>
              <parameter expression="${ctx:PROP_CODTRANSMISION}" type="CHAR"/>
              <parameter type="INTEGER" value="12"/>
              <parameter type="VARCHAR" value="SERV_REINTENTO_CONSULTA"/>
            </statement>
          </dbreport>
        </case>
      </switch>
      <drop description="End flow"/>
    </default>
  </switch>

```



```

</switch>
<switch source="$ctx:PROP_CODTRANSMISION">
  <case regex="0121|0122">
    <log description="IRM">
      <property name="LOG" value="+++++++ IRM +++++++"/>
    </log>
    <propertyGroup description="Tipo">
      <property name="PROP_TIPOTRANSMISION" scope="default" type="STRING" value="IRM"/>
    </propertyGroup>
    <sequence key="ReintentoConsultaSunatSeq"/>
    <drop description="End flow"/>
  </case>
  <case regex="RSA">
    <log description="RSA">
      <property name="LOG" value="+++++++ RSA +++++++"/>
    </log>
    <propertyGroup description="Tipo">
      <property name="PROP_TIPOTRANSMISION" scope="default" type="STRING" value="RSA"/>
    </propertyGroup>
    <sequence key="ReintentoConsultaSunatSeq"/>
    <drop description="End flow"/>
  </case>
  <default>
    <drop description="End flow"/>
  </default>
</switch>
</sequence>

```

Tabla 36. Código de flujo de proceso ReintentoConsultaMainErrorSeq

```

<sequence name="ReintentoConsultaMainErrorSeq" trace="disable" xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ REINTENTO CONSULTA (MAIN ERROR SEQUENCE)
+++++++"/>
    <property expression="get-property('ERROR_CODE')" name="ERROR_CODE"/>
    <property expression="get-property('ERROR_MESSAGE')" name="ERROR_MESSAGE"/>
    <property expression="get-property('ERROR_DETAIL')" name="ERROR_DETAIL"/>
    <property expression="get-property('ERROR_EXCEPTION')" name="ERROR_EXCEPTION"/>
  </log>
  <drop description="End flow"/>
</sequence>

```

Tabla 37. Código de flujo de proceso ReintentoConsultaSunatSeq

```

<sequence name="ReintentoConsultaSunatSeq" onError="ReintentoConsultaSunatErrorSeq" trace="disable"
xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log properties">
    <property name="LOG" value="+++++++ REINTENTO CONSULTA SUNAT (SEQUENCE) +++++++"/>
    <property expression="$ctx:PROP_TIPOTRANSMISION" name="TIPO_TRANSMISION"/>
    <property expression="$ctx:PROP_NUMTRANSMISION" name="NUMERO_TRANSMISION"/>
    <property expression="$ctx:PROP_CODTRANSMISION" name="CODIGO_TRANSMISION"/>
    <property expression="$ctx:PROP_TICKETENVIO" name="TICKET_ENVIO"/>
    <property expression="$ctx:PROP_ANHOENVIO" name="ANHO_ENVIO"/>
    <property expression="$ctx:PROP_IDTICKET" name="ID_TICKET"/>
    <property expression="$ctx:PROP_IDENVIOADUANA" name="ID_ENVIOADUANA"/>
  </log>
  <payloadFactory description="Create payload" media-type="xml">
    <format>

```

```

<soapenv:Envelope xmlns:ser="http://services.sigad.sunat.gob.pe"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <realizarConsulta xmlns="http://services.sigad.sunat.gob.pe">
      <parametrosConsulta
xmlns=""><![CDATA[<consulta><tipo>1</tipo><parametros><numeroTicket>$3</numeroTicket><annoTicket>$4
</annoTicket></parametros></consulta>]]></parametrosConsulta>
      </realizarConsulta>
    </soapenv:Body>
  </soapenv:Envelope>
</format>
<args>
  <arg evaluator="xml" expression="$ctx:PROP_TICKETENVIO"/>
  <arg evaluator="xml" expression="$ctx:PROP_ANHOENVIO"/>
</args>
</payloadFactory>
<log description="Log request payload" level="full"/>
<property description="Petición consulta" expression="/" name="PROP_PETICIONCONSULTA"
scope="default" type="STRING"/>
<property description="Content type" name="ContentType" scope="axis2" type="STRING" value="text/xml"/>
<call>
  <endpoint>
    <wsdl port="ReceptorWebServiceServiceImplPort" service="ReceptorService.htm"
uri="$SYSTEM:WSDL_URI">
      </wsdl>
    </endpoint>
  </call>
<log description="Log response payload" level="full"/>
<propertyGroup description="Respuesta consulta">
  <property expression="/" name="PROP_RESPUESTACONSULTA" scope="default" type="STRING"/>
  <property expression="/*[local-name()='realizarConsultaResultado']/text()"
name="PROP_RESPUESTAXMLBASE64" scope="default" type="STRING"/>
</propertyGroup>
<class description="Transform response"
name="pe.com.tpp.mediators.reintentoconsulta.TransformResponse">
  <property expression="$ctx:PROP_RESPUESTAXMLBASE64" name="encodedResponse"/>
</class>
<dbreport description="Update monitoreo">
  <connection>
    <pool>
      <driver>org.postgresql.Driver</driver>
      <uri>$SYSTEM:DB_PG_URL</uri>
      <user>$SYSTEM:DB_PG_USR</user>
      <password>$SYSTEM:DB_PG_PSW</password>
    </pool>
  </connection>
  <statement>
    <sql><![CDATA[UPDATE tbl_sda_monitoreo SET cpcrespuestaconsultaxml = ?, cpncodrespuesta = ?,
cpcinformensajerespuesta = ?, cpdfecharespuesta = CURRENT_TIMESTAMP, cpcrespuestaconsultaxml = ? WHERE
cpnnumerotransmision = ? AND cpccodtransmision = ? AND cpcticketenvio = ?]]></sql>
    <parameter expression="$ctx:PROP_RESPUESTACONSULTA" type="VARCHAR"/>
    <parameter expression="$ctx:statusCode" type="INTEGER"/>
    <parameter expression="$ctx:informacionMensaje" type="VARCHAR"/>
    <parameter expression="$ctx:responseXML" type="VARCHAR"/>
    <parameter expression="$ctx:PROP_NUMTRANSMISION" type="INTEGER"/>
    <parameter expression="$ctx:PROP_CODTRANSMISION" type="CHAR"/>
    <parameter expression="$ctx:PROP_TICKETENVIO" type="VARCHAR"/>
  </statement>
</dbreport>
<switch source="$ctx:PROP_EVENTOID">
  <case regex="13/9/11">
    <log description="Case 13/9/11">
      <property name="LOG" value="+++++++ EVENTO-ID: 13/9/11 +++++++"/>
    </log>
  </case>
</switch>

```

```

</log>
</case>
<case regex="7">
  <log description="Case 7">
    <property name="LOG" value="+++++++ EVENTO-ID: 7 +++++++"/>
  </log>
  <dbreport description="Insert control">
    <connection>
      <pool>
        <driver>org.postgresql.Driver</driver>
        <url>${SYSTEM:DB_PG_URL}</url>
        <user>${SYSTEM:DB_PG_USR}</user>
        <password>${SYSTEM:DB_PG_PSW}</password>
      </pool>
    </connection>
    <statement>
      <sql><![CDATA[INSERT INTO tbl_sda_control (cpnnumerotransmision, cpccodtransmision,
cpnid_evento, cpncodrespuesta, cpcinfomensajerespuesta, cpcrespuestaxml, cpcusuario_servicio) VALUES (?, ?,
?, ?, ?, ?, ?)]]></sql>
      <parameter expression="${ctx:PROP_NUMTRANSMISION}" type="INTEGER"/>
      <parameter expression="${ctx:PROP_CODTRANSMISION}" type="CHAR"/>
      <parameter expression="${ctx:PROP_EVENTOID}" type="INTEGER"/>
      <parameter expression="${ctx:statusCode}" type="INTEGER"/>
      <parameter expression="${ctx:informacionMensaje}" type="VARCHAR"/>
      <parameter expression="${ctx:responseXML}" type="VARCHAR"/>
      <parameter type="VARCHAR" value="SERV_CONSULTA"/>
    </statement>
  </dbreport>
  <drop description="End flow"/>
</case>
<default>
  <log description="Default">
    <property name="LOG" value="+++++++ EVENTO-ID: 10 (REINTENTAR CONSULTA) +++++++"/>
  </log>
  <dbreport description="Insert control">
    <connection>
      <pool>
        <driver>org.postgresql.Driver</driver>
        <url>${SYSTEM:DB_PG_URL}</url>
        <user>${SYSTEM:DB_PG_USR}</user>
        <password>${SYSTEM:DB_PG_PSW}</password>
      </pool>
    </connection>
    <statement>
      <sql><![CDATA[INSERT INTO tbl_sda_control (cpnnumerotransmision, cpccodtransmision,
cpnid_evento, cpncodrespuesta, cpcinfomensajerespuesta, cpcrespuestaxml, cpcusuario_servicio) VALUES (?, ?,
?, ?, ?, ?, ?)]]></sql>
      <parameter expression="${ctx:PROP_NUMTRANSMISION}" type="INTEGER"/>
      <parameter expression="${ctx:PROP_CODTRANSMISION}" type="CHAR"/>
      <parameter expression="${ctx:PROP_EVENTOID}" type="INTEGER"/>
      <parameter expression="${ctx:statusCode}" type="INTEGER"/>
      <parameter expression="${ctx:informacionMensaje}" type="VARCHAR"/>
      <parameter expression="${ctx:responseXML}" type="VARCHAR"/>
      <parameter type="VARCHAR" value="SERV_REINTENTO_CONSULTA"/>
    </statement>
  </dbreport>
  <sequence key="ReintentoConsultaSunatErrorSeq"/>
  <drop description="End flow"/>
</default>
</switch>
<dblookup description="Actualizar minutos">
  <connection>
    <pool>

```

```

    <driver>org.postgresql.Driver</driver>
    <url>${SYSTEM:DB_PG_URL}</url>
    <user>${SYSTEM:DB_PG_USR}</user>
    <password>${SYSTEM:DB_PG_PSW}</password>
  </pool>
</connection>
<statement>
  <sql><![CDATA[SELECT funsda_actualizarminutos(?, ?, ?)]></sql>
  <parameter expression="${ctx:PROP_NUMTRANSMISION}" type="INTEGER"/>
  <parameter expression="${ctx:PROP_CODTRANSMISION}" type="VARCHAR"/>
  <parameter expression="${ctx:PROP_TIPOTRANSMISION}" type="VARCHAR"/>
</statement>
</dblookup>
<log description="Send response to on-premise">
  <property name="LOG" value="+++++++ ENVIAR RESPUESTA DE SUNAT A ON-PREMISE +++++++"/>
</log>
<payloadFactory description="Enviar respuesta" media-type="json">
  <format>
    {"numero_transmision":"$1", "codigo_transmision":"$2", "codigo_respuesta":"$3",
"mensaje_respuesta":"$4", "id_ticket":"$5", "id_envioaduana":"$6", "respuesta_xml":"$7"}</format>
  <args>
    <arg evaluator="xml" expression="${ctx:PROP_NUMTRANSMISION}"/>
    <arg evaluator="xml" expression="${ctx:PROP_CODTRANSMISION}"/>
    <arg evaluator="xml" expression="${ctx:statusCode}"/>
    <arg evaluator="xml" expression="${ctx:informacionMensaje}"/>
    <arg evaluator="xml" expression="${ctx:PROP_IDTICKET}"/>
    <arg evaluator="xml" expression="${ctx:PROP_IDENVIADUANA}"/>
    <arg evaluator="xml" expression="${ctx:PROP_RESPUESTAXMLBASE64}"/>
  </args>
</payloadFactory>
<log description="Log json payload" level="full"/>
<property description="Content type" name="ContentType" scope="axis2" type="STRING"
value="application/json"/>
<script description="Environment variables"
language="js"><![CDATA[mc.setProperty("PROP_KAFKA_BOOTSTRAP_SERVERS",
java.lang.System.getenv("KAFKA_BOOTSTRAP_SERVERS"));]]></script>
<kafkaTransport.init>
  <bootstrapServers>${ctx:PROP_KAFKA_BOOTSTRAP_SERVERS}</bootstrapServers>
</kafkaTransport.init>
<kafkaTransport.publishMessages>
  <topic>tpp_transmision_resultado</topic>
</kafkaTransport.publishMessages>
<log description="Log end">
  <property name="LOG" value="+++++++ FIN DE TRANSMISION +++++++"/>
</log>
<drop description="End transmision"/>
</sequence>

```

Tabla 38. Código de flujo de proceso ReintentoConsultaSunatErrorSeq

```

<sequence name="ReintentoConsultaMainErrorSeq" trace="disable" xmlns="http://ws.apache.org/ns/synapse">
  <log description="Log sequence">
    <property name="LOG" value="+++++++ REINTENTO CONSULTA (MAIN ERROR SEQUENCE)
+++++++"/>
    <property expression="get-property('ERROR_CODE')" name="ERROR_CODE"/>
    <property expression="get-property('ERROR_MESSAGE')" name="ERROR_MESSAGE"/>
    <property expression="get-property('ERROR_DETAIL')" name="ERROR_DETAIL"/>
    <property expression="get-property('ERROR_EXCEPTION')" name="ERROR_EXCEPTION"/>
  </log>
  <drop description="End flow"/>
</sequence>

```

h.- Sprint 3.

Como primera actividad se programó la reunión de Sprint Planning con los miembros del equipo básico scrum.

El Scrum master y el Equipo Scrum definieron las tareas del Sprint backlog para el Sprint 3.

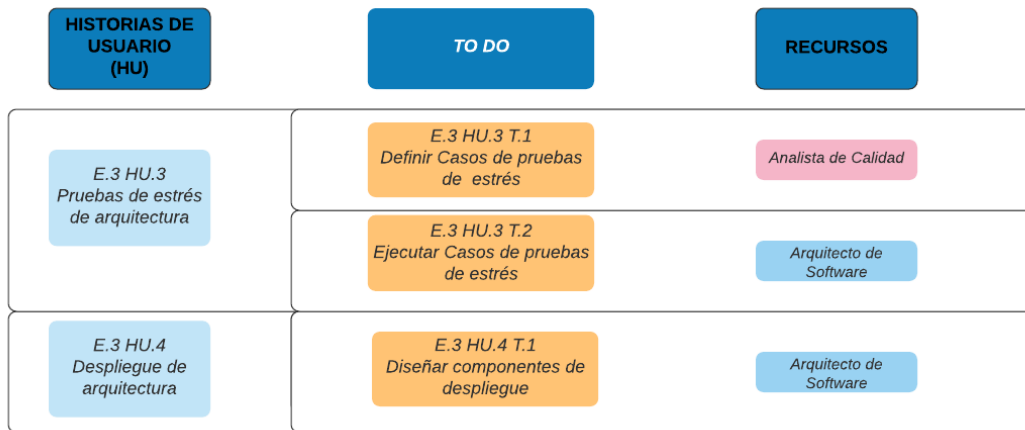


Figura 25-Sprint Backlog del Sprint 03 (Fuente: Elaboración Propia)

Ejecución de tareas.

Definir casos de pruebas de estrés

La definición de los casos de pruebas y de las métricas fueron responsabilidad del Analista de Calidad de TPP. Definió los casos de pruebas (Ver tabla 35).

El objetivo de las pruebas de estrés es identificar el límite de escalabilidad ante un incremento de peticiones de transmisiones en un tiempo reducido.

Las siguientes fueron las métricas de pruebas de estrés:

Tabla 39. Métricas de pruebas de estrés

Métrica	Descripción
Número de Hilos	Cantidad de hilos en las que se va a mandar la petición
Periodo de subida	Tiempo en segundos que se ejecutarán el número de hilos
Contador bucle	Cantidad de veces que se va a repetir el proceso

Tabla 40. Casos de pruebas de estrés

Métrica	Descripción
Caso de prueba 1	Cantidad de hilos: 50 Periodo de Subida: 60 seg Contador bucle:1
Caso de prueba 2	Cantidad de hilos: 50

	Periodo de Subida: 60 seg Contador bucle:2
--	---

Ejecutar casos de pruebas de estrés

La ejecución de los casos de pruebas fueron responsabilidad del Arquitecto de Software. Para su ejecución se usó la herramienta JMeter que permite hacer peticiones considerando las métricas definidas.

Caso de Prueba 1:

Cantidad de hilos: 50

Periodo de Subida: 60

Contador bucle:1

Se define los elementos que permitirán realizar la petición del caso de prueba.

- Cliente de colas (JMS Punto a Punto) para el envío del documento de transmisión en formato JSON. El cliente hace la petición a cola “source.tpp_transmission”

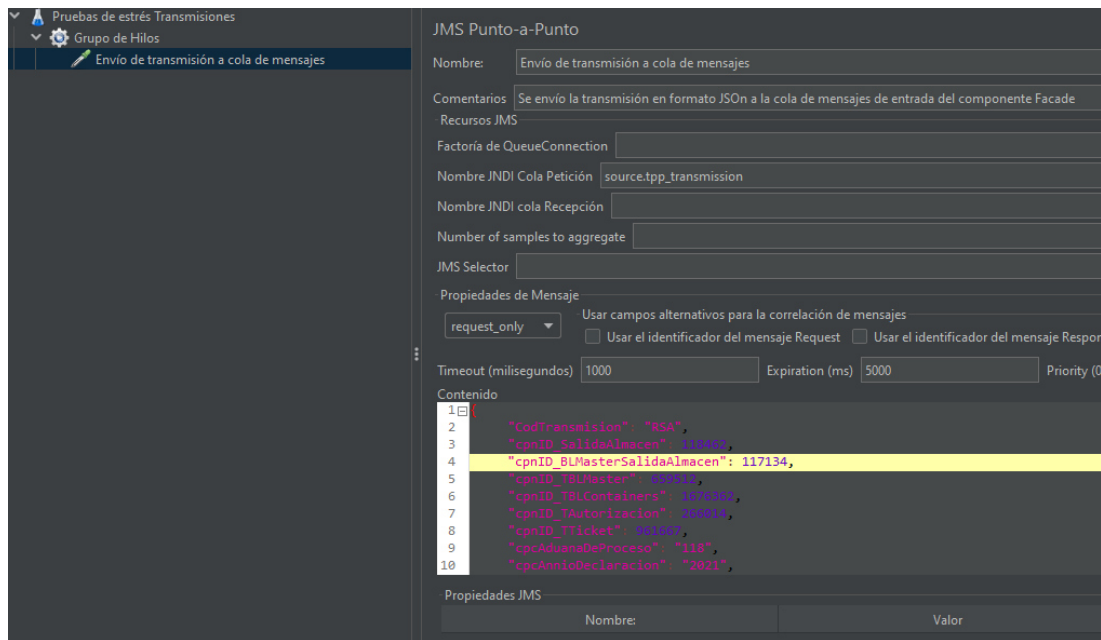


Figura 26-Cliente de petición de prueba de Caso 1 (Fuente: Elaboración Propia)

- Grupo de Hilos para la configuración de las métricas definidas para la prueba.

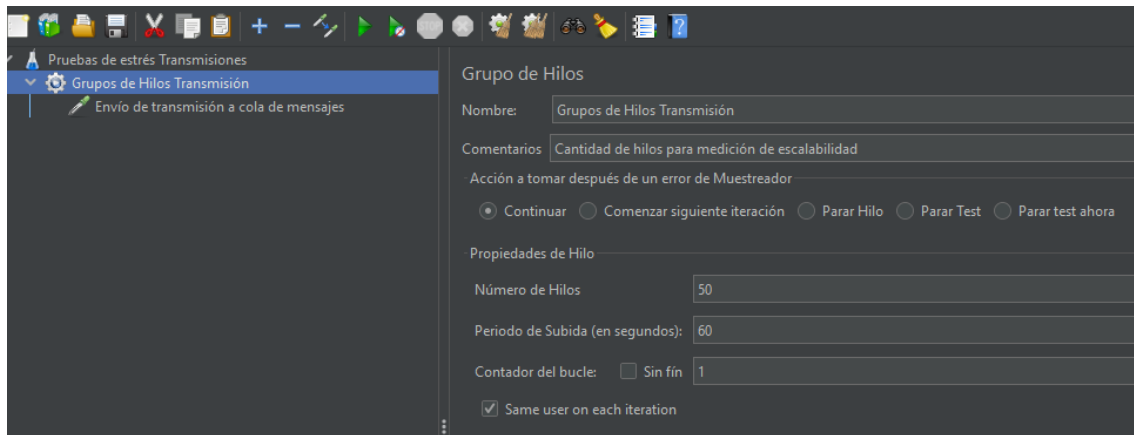


Figura 27-Configuración de métricas de prueba de Caso 1 (Fuente: Elaboración Propia)

- Se ejecuta el caso de prueba.
Obteniendo como resultado la ejecución exitosa (verde) de 50 hilos durante 60 segundos en un solo bucle.

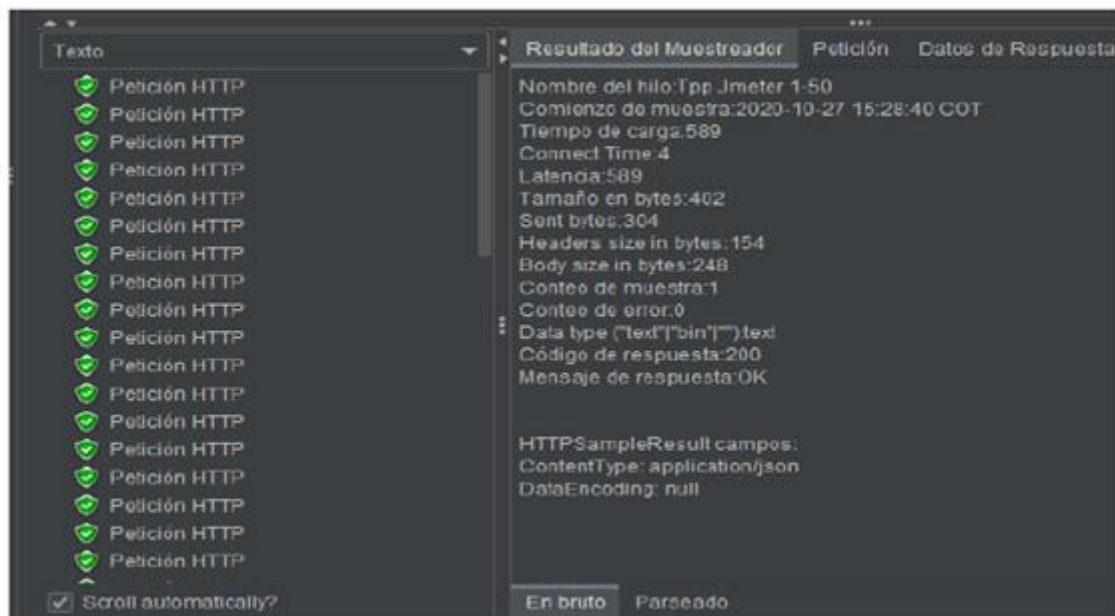


Figura 28-Resultado de peticiones de prueba de Caso 1 (Fuente: Elaboración Propia)

- El resumen del reporte del caso de prueba.
Total de errores: 0%
Tiempo de respuesta mínimo: 532 milisegundos
Tiempo de respuesta máximo: 810 milisegundos
Tiempo promedio de respuesta: 601 milisegundos

Reporte resumen

Nombre:

Comentarios:

Escribir todos los datos a Archivo

Nombre de archivo: Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Byt...
Petición HT...	50	601	532	810	48.66	0.00%	10.2/min	0.07	0.05	402.0
Total	50	601	532	810	48.66	0.00%	10.2/min	0.07	0.05	402.0

Figura 29-Resumen estadístico de prueba de Caso 1 (Fuente: Elaboración Propia)

Caso de Prueba 2:

Cantidad de hilos: 50
 Periodo de Subida: 60
 Contador bucle:2

Se define los elementos que permitirán realizar la petición del caso de prueba.

- Cliente de colas (JMS Punto a Punto) para el envío del documento de transmisión en formato JSON. El cliente hace la petición a cola "source.tpp_transmission"

JMS Punto-a-Punto

Nombre:

Comentarios:

Recursos JMS

Factoría de QueueConnection:

Nombre JNDI Cola Petición:

Nombre JNDI cola Recepción:

Number of samples to aggregate:

JMS Selector:

Propiedades de Mensaje

Usar campos alternativos para la correlación de mensajes

Usar el identificador del mensaje Request Usar el identificador del mensaje Respon

Timeout (milisegundos): Expiration (ms): Priority:

Contenido

```

1
2
3
4
5
6
7
8
9
10
"codTransmision": "RSA",
"cpnID_SalidaAlmacen": "118462",
"cpnID_BLMasterSalidaAlmacen": "117134",
"cpnID_BLMaster": "117134",
"cpnID_TRLContainers": "1878382",
"cpnID_TAutorizacion": "266814",
"cpnID_TTicket": "848897",
"cocAduanaDeProceso": "118",
"cocAnnoDeDeclaracion": "2021",

```

Propiedades JMS

Nombre:	Valor
---------	-------

Figura 30-Cliente de petición de prueba de Caso 2 (Fuente: Elaboración Propia)

- Grupo de Hilos para la configuración de las métricas definidas para la prueba.

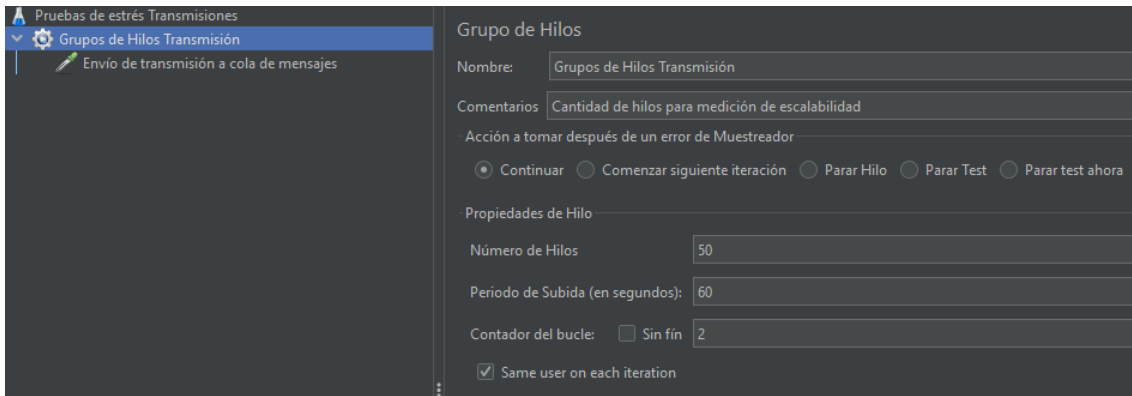


Figura 31-Configuración de métricas de prueba de Caso 2 (Fuente: Elaboración Propia)

- Se ejecuta el caso de prueba.
Obteniendo como resultado la ejecución exitosa (verde) de 50 hilos durante 100 segundos en un solo bucle.

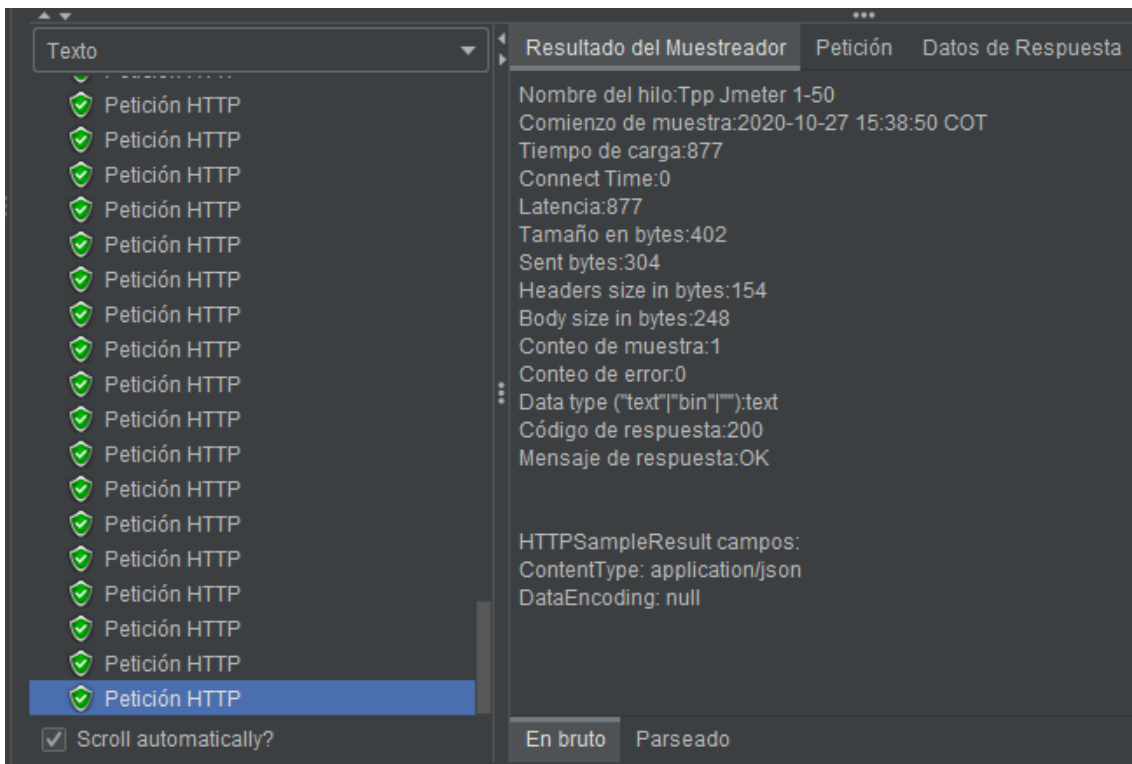


Figura 32-Resultado de peticiones de prueba de Caso 2 (Fuente: Elaboración Propia)

- El resumen del reporte del caso de prueba.
Total de errores: 0%
Tiempo de respuesta mínimo: 536 milisegundos
Tiempo de respuesta máximo: 1410 milisegundos
Tiempo promedio de respuesta: 755 milisegundos

Reporte resumen

Nombre:

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Byt...
Petición HT...	100	755	536	1410	132.07	0.00%	20.3/min	0.13	0.10	402.0
Total	100	755	536	1410	132.07	0.00%	20.3/min	0.13	0.10	402.0

Figura 33-Resumen estadístico de prueba de Caso 2 (Fuente: Elaboración Propia)

Diseñar componentes de despliegue

El estudio y análisis del sizing de los componentes físicos estuvo a cargo del Arquitecto de Software. Se consideraron los mejores recursos priorizando la escalabilidad de la arquitectura a un bajo costo por demanda.

Se definió los siguientes componentes para la ejecución en un entorno en nube (Amazon Web Services):

1.- EC2 Large (1 Instancia).

Características:

Nombre	CPUs virtual	Memoria (GiB)	Almacenamiento de instancias	Ancho de banda de red (Gbps)	Ancho de banda de EBS (Mbps)
c6g.xlarge	4	8	Solo EBS	Hasta 10	Hasta 4750

1.1.- Docker

Docker se despliegue en EC2 Large (1 Instancia)			
Consola administrativa	Conectividad Privada	Cifrado y Seguridad	Escalabilidad
Amazon EKS	Amazon VPC	Autenticador IAM	Escala automatizada por amazon EKS

2.- EC2 Medium (2 Instancias).

Características:

Nombre	CPUs virtual	Memoria (GiB)	Almacenamiento de instancias	Ancho de banda de red (Gbps)	Ancho de banda de EBS (Mbps)
c6g.medium	1	2	Solo EBS	Hasta 10	Hasta 4750

2.1.- Amazon RDS

Amazon RDS se despliegue en EC2 Medium (1 Instancia)			
Disponibilidad		Escalabilidad	
Copia de Seguridad automática	Sustitución automática de hospedaje	Escalado de Informática	Almacenamiento de instancias
Maximo 35 dias	Si, en caso de error de hardware	Máximo de 32 vCPU y 244 GiB de RAM	Máximo de 64 TB

2.2.- Amazon Kafka

Amazon Kafka se despliegue en EC2 Medium (1 Instancia)			
Consola administrativa	Conectividad Privada	Cifrado y Seguridad	Escalabilidad
Amazon MSK	Amazon VPC	AWS Key ManagementService (KMS)	Hasta 15 nodos por clúster

Se diseñó el siguiente diagrama de despliegue de componentes integrando los componentes físicos y de software.

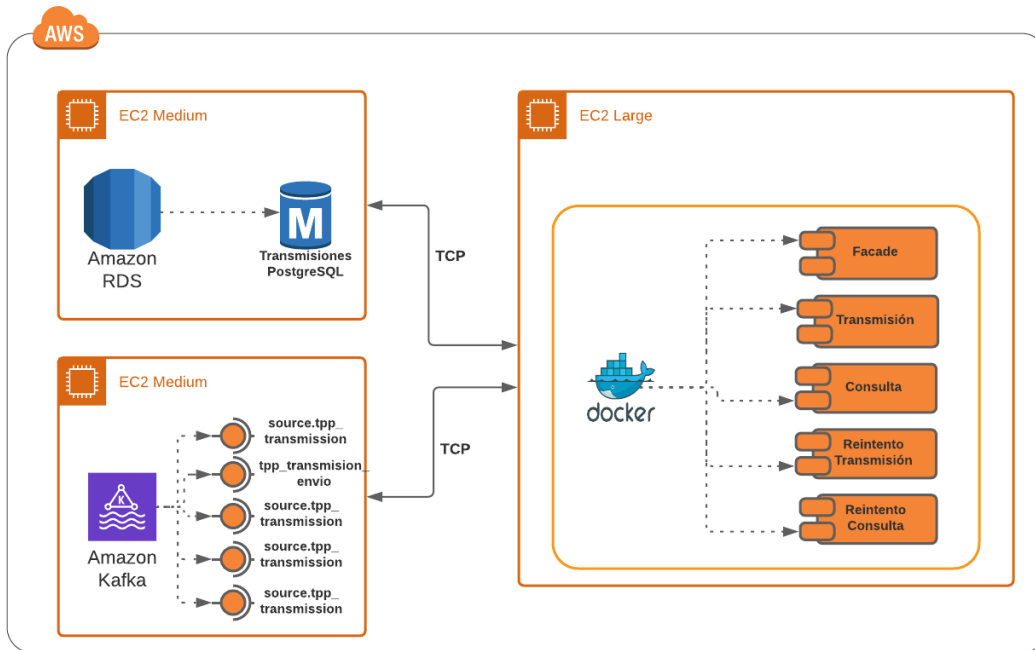


Figura 34-Diagrama de despliegue de componentes de arquitectura (Fuente: Elaboración Propia)

i.- Sprint 4.

Como primera actividad se programó la reunión de Sprint Planning con los miembros del equipo básico scrum.

El Scrum master y el Equipo Scrum definieron las tareas del Sprint backlog para el Sprint 4.

HISTORIAS DE USUARIO (HU)	TO DO	RECURSOS
E.4 HU.1 Desarrollar componentes Backend	E.4 HU.1 T.1 Desarrollar componentes Backend de monitoreo	Desarrollador Backend
E.4 HU.2 Desarrollar componentes Frontend	E.4 HU.2 T.1 Diseñar componentes Frontend de monitoreo	Desarrollador Frontend

Figura 35-Sprint Backlog del Sprint 04 (Fuente: Elaboración Propia)

Ejecución de tareas.

Desarrollar componentes Backend de monitoreo

El desarrollo de los componentes backend para el monitoreo de las transmisiones fue responsabilidad del Desarrollador Backend.

Los componentes backend son los siguientes:

IRM API: Métodos para consulta de datos, estados y respuesta para el monitoreo de transmisiones IRM.

RSA API: Métodos para consulta de datos, estados y respuesta para el monitoreo de transmisiones RSA.

Amazon API Gateway: Capa de seguridad que permite filtrar las peticiones a los API de monitoreo.

Transmisiones PostgreSQL: Base de datos para el monitoreo de Transmisiones en PostgreSQL.

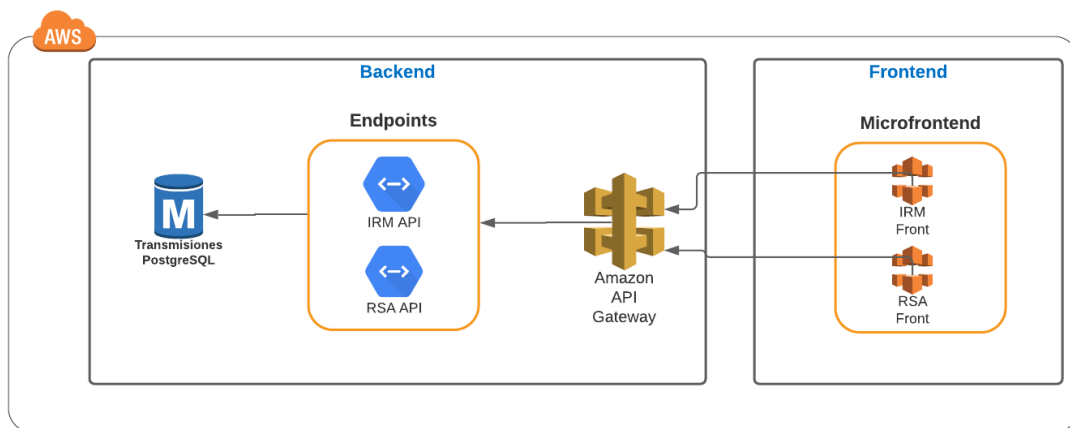


Figura 36-Diagrama de componentes backend (Fuente: Elaboración Propia)

Desarrollar componentes Frontend de monitoreo

El desarrollo de los componentes frontend para el monitoreo de las transmisiones fue responsabilidad del Desarrollador Frontend.

Se desarrolló las siguientes pantallas para IRM:

IRM

Gráfico
Reporte
Actualizar
Carga:
Local:
Filtrar

Última actualización: 29/04/2021 09:26:12 PM

Bill of Landing	Recalada	Fecha y Hora Último Ingreso	Tipo Transmisión	T. Transcurrido	T. Restante	Estado de Transmisión	Mensaje Error	Acción
ONEYZHOA37064500	AGAMEMNON 1409	15/04 03:10 PM	NUMERACIÓN	342:16 hrs	00:00 hrs	Envío rechazado	EL MANIFIESTO 01-IIB...	...
L9930034581	AGAMEMNON 1410	15/04 12:58 PM	NUMERACIÓN	344:27 hrs	00:00 hrs	Envío rechazado	EL MANIFIESTO 01-IIB...	...
ONEYZHOA37064500	AGAMEMNON 1409	15/04 03:00 AM	NUMERACIÓN	354:26 hrs	00:00 hrs	Envío rechazado	EL MANIFIESTO 01-IIB...	...
ONEYZHOA37064500	AGAMEMNON 1409	15/04 03:00 AM	NUMERACIÓN	354:26 hrs	00:00 hrs	Envío rechazado	EL MANIFIESTO 01-IIB...	...
ONEYZHOA37064500	AGAMEMNON 1409	15/04 03:00 AM	RECTIFICACIÓN	354:26 hrs	00:00 hrs	Envío rechazado	EL MANIFIESTO 01-IIB...	...
ONEYSUBA39613900	AGAMEMNON 1409	14/04 10:00 AM	NUMERACIÓN	371:26 hrs	00:00 hrs	Envío rechazado	EL MANIFIESTO 01-IIB...	...
L9930034582	AGAMEMNON 1410	05/04 08:00 PM	NUMERACIÓN	577:26 hrs	00:00 hrs	Error de comunicación	-	...
L9930034582	AGAMEMNON 1410	05/04 08:00 PM	NUMERACIÓN	577:26 hrs	00:00 hrs	Reintentando transmisión	-	...
L9930034581	AGAMEMNON 1410	05/04 11:27 AM	NUMERACIÓN	585:58 hrs	00:00 hrs	Reintentando transmisión	-	...

Figura 37-Pantalla de cabecera de monitoreo de IRM (Fuente: Elaboración Propia)

IRM

Gráfico
Reporte
Actualizar
Carga:
Local:
Filtrar

Última actualización: 29/04/2021 09:26:12 PM

Bill of Landing	Recalada	Fecha y Hora Último Ingreso	Tipo Transmisión	T. Transcurrido	T. Restante	Estado de Transmisión	Mensaje Error	Acción
ONEYZHOA37064500	AGAMEMNON 1409	15/04 03:10 PM	NUMERACIÓN	342:16 hrs	00:00 hrs	Envío rechazado	EL MANIFIESTO 01-IIB...	...
Bill of Landing: ONEYZHOA37064500 Detalle: 1275			Manifiesto: 2021-00090 Condición carga: Contenedor FULL			Cant. Bultos: 2200 Peso Bruto: 5430		
L9930034581	AGAMEMNON 1410	15/04 12:58 PM	NUMERACIÓN	344:27 hrs	00:00 hrs	Envío rechazado	EL MANIFIESTO 01-IIB...	...
ONEYZHOA37064500	AGAMEMNON 1409	15/04 03:00 AM	NUMERACIÓN	354:26 hrs	00:00 hrs	Envío rechazado	EL MANIFIESTO 01-IIB...	...
ONEYZHOA37064500	AGAMEMNON 1409	15/04 03:00 AM	NUMERACIÓN	354:26 hrs	00:00 hrs	Envío rechazado	EL MANIFIESTO 01-IIB...	...
ONEYZHOA37064500	AGAMEMNON 1409	15/04 03:00 AM	RECTIFICACIÓN	354:26 hrs	00:00 hrs	Envío rechazado	EL MANIFIESTO 01-IIB...	...
ONEYSUBA39613900	AGAMEMNON 1409	14/04 10:00 AM	NUMERACIÓN	371:26 hrs	00:00 hrs	Envío rechazado	EL MANIFIESTO 01-IIB...	...
L9930034582	AGAMEMNON 1410	05/04 08:00 PM	NUMERACIÓN	577:26 hrs	00:00 hrs	Error de comunicación	-	...

Figura 38-Pantalla de detalle de monitoreo de IRM (Fuente: Elaboración Propia)

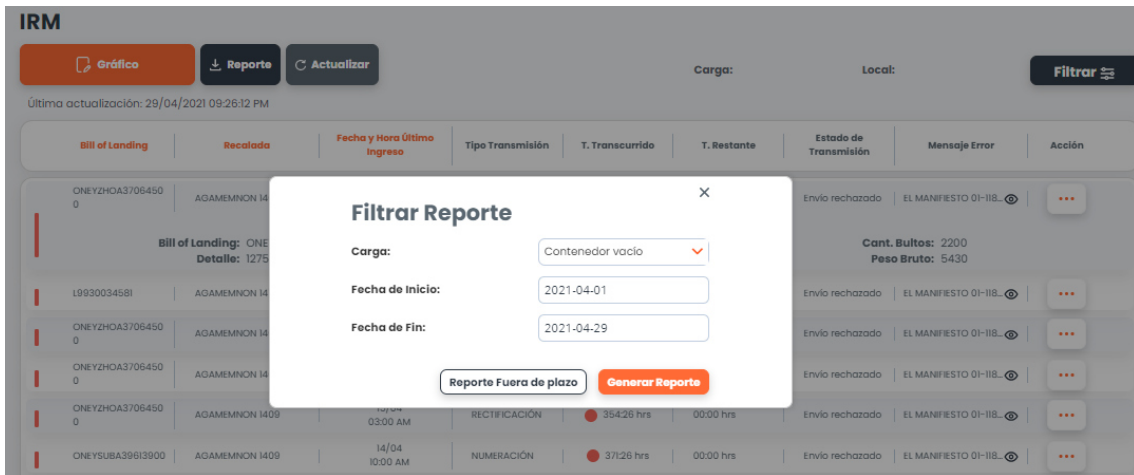


Figura 39-Pantalla de reporte de monitoreo de IRM (Fuente: Elaboración Propia)



Figura 40-Pantalla de gráfico estadístico de monitoreo de IRM (Fuente: Elaboración Propia)

Se desarrolló las siguientes pantallas para RSA.

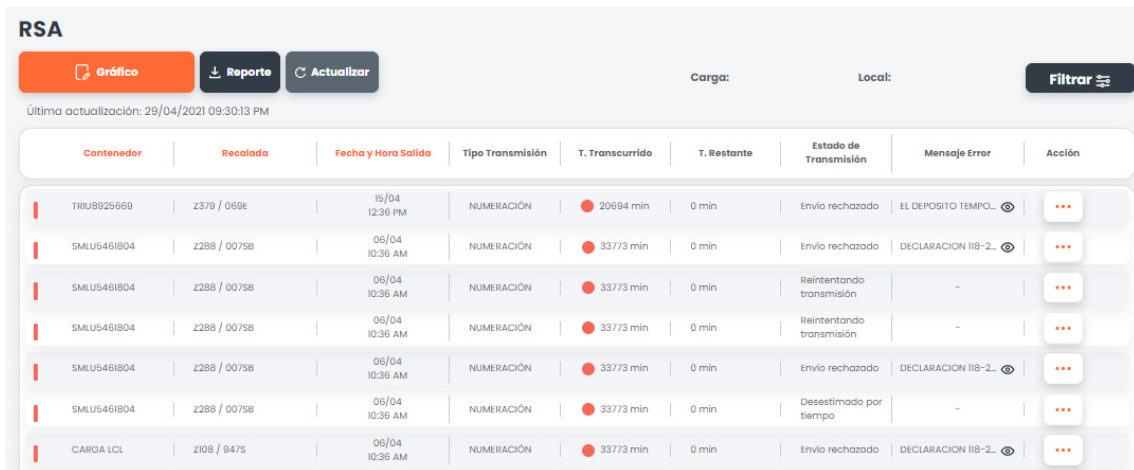


Figura 41-Pantalla de cabecera de monitoreo de RSA (Fuente: Elaboración Propia)

RSA

Gráfico | Reporte | Actualizar

Carga: Local: Filtrar

Última actualización: 29/04/2021 09:30:33 PM

Contenedor	Recalada	Fecha y Hora Salida	Tipo Transmisión	T. Transcurrido	T. Restante	Estado de Transmisión	Mensaje Error	Acción
TRIU8925669	Z379 / 069E	15/04 12:36 PM	NUMERACIÓN	20694 min	0 min	Envío rechazado	EL DEPOSITO TEMP...	...
Bill of Lading: MEDUVCI44288 Tipo Carga: FCL Ticket Puerto: 840569 Nro. Volante: 181541 Tara Camión: 4000			Manifiesto: 2020-02813 Cant. Bultos: 975 Placa: ACN873 Nro. DAM: 118-2021-10-1550 Peso Bruto: 10000 Régimen: 10			Detalle: 1542 Aut. Retiro: 2100000165 Brevete: Q21130271 RUC Transportista: 20486728338 Peso Neto: 6000		
SMLU5461804	Z288 / 0075B	06/04 10:36 AM	NUMERACIÓN	33773 min	0 min	Envío rechazado	DECLARACION I18-2...	...
SMLU5461804	Z288 / 0075B	06/04 10:36 AM	NUMERACIÓN	33773 min	0 min	Reintentando transmisión	-	...
SMLU5461804	Z288 / 0075B	06/04 10:36 AM	NUMERACIÓN	33773 min	0 min	Reintentando transmisión	-	...

Figura 42-Pantalla de detalle de monitoreo de RSA (Fuente: Elaboración Propia)

RSA

Gráfico | Reporte | Actualizar

Carga: Local:

Última actualización: 11/03/2021 04:31:11 PM

Contenedor	Recalada	Fecha y Hora Salida	Tipo Transmisión	T. Transcurrido	T. Restante	Estado de Transmisión	Mensaje Error
GESU9342159	Z379 / 069E	09/03 12:29 PM	NUMERACIÓN	3125 min	-3096 min	Envío rechazado	EL DEPOSITO TEMP...
TRIU8925669	Z379 / 069E	09/03	NUMERACIÓN	4291 min	-4262 min	Envío rechazado	EL DEPOSITO TEMP...
WHLU5614147	SUPB / 02					Envío rechazado	

Filtrar Reporte

Carga: 5 cargos seleccionadas

Fecha de Inicio: 2021-03-10

Fecha de Fin: 2021-03-11

Reporte Fuera de plazo | Generar Reporte

Figura 43-Pantalla de reporte de monitoreo de RSA (Fuente: Elaboración Propia)

3.2.5 FUNDAMENTOS UTILIZADOS

3.2.5.1 MARCO TEORICO

Los siguientes son los fundamentos utilizados en el presente informe para la implementación del proyecto.

Transmisiones Aduaneras.

Son las actividades relacionadas a la transmisión, registro y sus documentos aduaneros, también relacionados con el ingreso y salida de las mercancías, los medios que realizan el transporte, con el objetivo de lograr el cumplimiento de las normas que lo regulan. (Fuente: SUNAT (2020). Manifiesto de Carga)

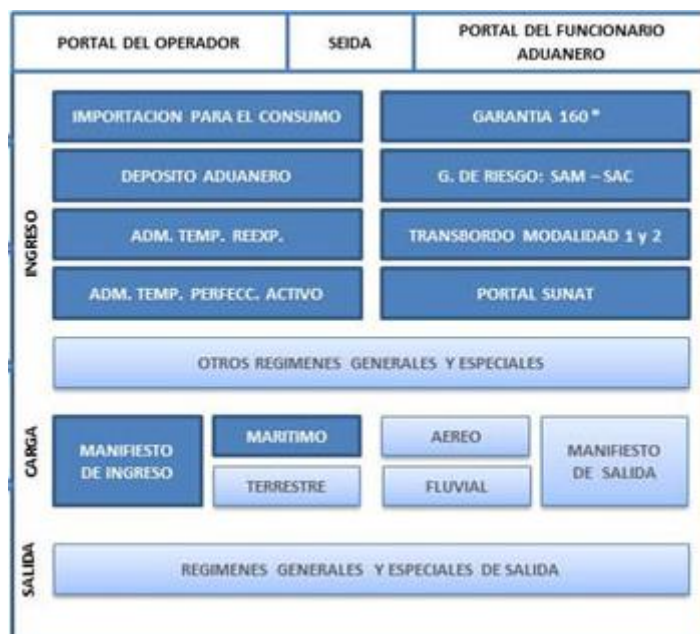


Figura 44-Sistema de Despacho Aduanero (Fuente: SUNAT (2020). Operatividad Aduanera)

Dirigido a los siguientes operadores y personal:

1. OCE: Operador de comercio exterior
 - a) transportista;
 - b) agente de carga internacional;
 - c) almacén aduanero;
 - d) despachador de aduana y
 - e) empresa de servicio de entrega rápida.

2. OI: Operador interviniente
 - a) concesionario de las instalaciones portuarias y aeroportuarias;
 - b) zona franca de Tacna;
 - c) zona especial de desarrollo;
 - d) terminal de carga del transportista aéreo;
 - e) consignatario y
 - f) operador de base fija.

3. Personal de SUNAT.

La responsabilidad del trámite aduanero corresponde a los siguientes cargos:

- Intendente Nacional de Control Aduanero, del Intendente Nacional de Sistemas de Información.
- Intendente Nacional de Desarrollo e Innovación Aduanera.

- Intendentes de aduana de la República,
- Jefaturas y del personal de las distintas unidades de organización que intervienen.

Los procedimientos para la transmisión de la información necesitan los siguientes pasos:

1. Regular la transmisión y presentación de información de:
El manifiesto de carga de ingresos o salidas, los documentos vinculados, los actos relacionados y los medios de transporte.
2. Medios de transmisión utilizados para el envío de información a SUNAT son: El OCE y el OI que transmiten la información a través de los sistemas informáticos de intercambio de datos electrónicos o manualmente a través de portal de la SUNAT.
El sistema informático valida la información transmitida y de ser conforme comunica la aceptación. Caso contrario se genera un mensaje de no aceptación.
3. La numeración del manifiesto de carga aduanera es generada correlativamente, en relación con el tipo de transporte, en cada intendencia de aduana.
La obligación de transmitir la información del manifiesto de carga se considera cumplida con la totalidad de los documentos de transporte que lo conforman.
4. El transportista transmite la información del manifiesto de carga en los plazos de acuerdo al tipo de transporte.
 - a) En la vía marítima:
 - Datos generales del medio de transporte, hasta ciento sesenta y ocho horas antes de la llegada del medio de transporte;
 - Información de los documentos de transporte que contienen carga contenerizada hasta 72 horas y de carga no contenerizada hasta 24 horas.
 - b) En la vía aérea:
 - Datos generales del medio de transporte, hasta 24 horas antes de la llegada del medio de transporte;
 - Información de los documentos de transporte, hasta 4 horas antes de la llegada del medio de transporte;
 - c) En el resto de vías, la información de los datos generales y sus documentos de transporte hasta antes de la llegada del medio de transporte.
5. Plazos de transmisión de los documentos vinculados: La transmisión de la información de los documentos vinculados se realiza hasta antes de la llegada del medio de

transporte. En la vía marítima, se da por cumplida la obligación con la transmisión del número del Término de Embarque.

6. Transmisión de documentos vinculados a través de la Casilla Electrónica del Usuario (CEU): Mientras no se implemente el sistema de información para la transmisión de los documentos vinculados, el transportista efectúa la transmisión de estos documentos desde su CEU a la Casilla Electrónica Corporativa Aduanera (CECA).

Metodología Scrum.

Inicialmente scrum fue desarrollado para gestionar y desarrollar productos. Desde inicios de los años 90 Scrum se ha usado ampliamente en todo el mundo para (Fuente: Ken Schwaber, Jeff SutherlandLa (2017). Guía de Scrum):

1. Investigar e identificar mercados viables, tecnologías y capacidades de productos;
2. Desarrollar productos y mejoras;
3. Liberar productos y mejoras tantas veces como sea posible durante el día;
4. Desarrollar y mantener ambientes en la Nube (en línea, seguros, bajo demanda) otros entornos operacionales para el uso de productos; y
5. Mantener y renovar productos.

Scrum se ha usado para desarrollar software, hardware, software embebido, redes de funciones interactivas, vehículos autónomos, escuelas, gobiernos, mercadeo, también para gestionar la operación de organizaciones y casi todo lo que usamos en nuestra vida diaria, como individuo y como sociedad.

Dado que la complejidad de la tecnología, el mercado y del entorno y sus interacciones aumentan rápidamente, la utilidad de Scrum para tratar con la complejidad está a prueba diariamente.

Scrum demostró ser especialmente efectivo en la transferencia iterativa e incremental de conocimiento. Scrum se usa ahora ampliamente para productos, servicios y gestión de la organización matriz.

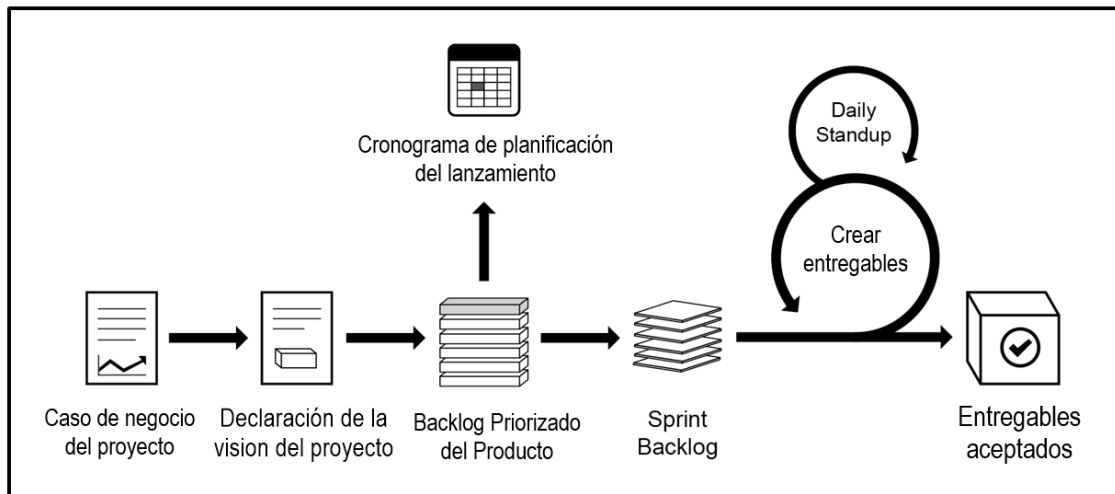


Figura 45-Flujo de Scrum para un Sprint (Fuente: Tridibesh Satpathy (2017). Guía para el Cuerpo de Conocimiento de Scrum, Tercera Edición)

Scrum asegura que el conocimiento procede de la experiencia y de tomar decisiones basadas en lo que ya se conoce. Es iterativo e incremental para mejorar la predicción y reducir el riesgo. Tres principales conceptos soportan a scrum: transparencia, inspección y adaptación.

A.-Transparencia.

La transparencia requiere que los aspectos significativos del proceso sean definidos por un estándar común, de tal forma que los observadores puedan entender la información que se comparte.

Por ejemplo:

- Los miembros del equipo básico scrum deben compartir un lenguaje común para referirse al proceso.

B.-Inspección.

Los miembros del equipo básico scrum deben poder inspeccionar el avance de los entregables y así detectar variaciones no deseadas. Se recomienda que no sea tan frecuente para no interferir en el trabajo del equipo.

C.-Adaptación.

Si de las inspecciones realizadas se determina que el producto resultante será no aceptable, se debe ajustar el proceso cuanto antes para reducir desviaciones.

Scrum formula cuatro eventos formales dentro del Sprint, para la inspección y adaptación.

- Sprint Planning: Planificación del Sprint
- Daily Scrum: Scrum Diario
- Sprint Review: Revisión del Sprint
- Sprint Retrospective: Retrospectiva del Sprint

El Equipo Básico Scrum (*Scrum Team*).

El Equipo Básico Scrum consiste en un Dueño de Producto (*Product Owner*), el Scrum Master y el Equipo de Desarrollo (*Development Team*). Es autoorganizado para elegir la mejor forma de ejecutar su trabajo, es multifuncional debido a que todos los miembros del equipo poseen diferentes habilidades técnicas.

El Dueño de Producto (*Product Owner*).

Es el responsable de maximizar el valor del producto resultado del trabajo del Equipo de Desarrollo y de gestionar la Lista del Producto (*Product Backlog*). La gestión de la Lista del Producto incluye:

- Definir claramente la Lista del Producto.
- Priorizar la Lista del Producto para cumplir los objetivos.
- Optimizar el valor resultado del trabajo del equipo de desarrollo;

El Equipo de Desarrollo (*Development Team*)

El Equipo de Desarrollo está conformado por profesionales que realizan las tareas planificadas al inicio de cada Sprint. Son los que generan el valor incremental para la revisión al final del Sprint. El trabajo en equipo fortalece la eficiencia y efectividad del equipo.

Tienen las siguientes características:

- Autoorganizados. El equipo define cómo la lista de productos genera funcionalidad desplegable de forma incremental.
- Multifuncionales. El equipo posee todas las habilidades necesarias para generar funcionalidad incremental.
- La responsabilidad siempre recae en todos los miembros del Equipo de Desarrollo.
- Es el único Equipo de Desarrollo responsable de la generación de valor, no se reconoce otro equipo subalterno.

Escalabilidad.

Es la capacidad que tiene un sistema de adaptarse a un crecimiento de carga sin perder calidad de los servicios ofrecidos (Fuente: Oscar Blancarte (2020). Introducción a la arquitectura de software- Un enfoque práctico, Primera Edición)

Existen dos maneras de escalar un sistema: Escalado vertical y escalado horizontal.

Escalado vertical (scale in), es cuando se incrementa los recursos en un solo componente físico, ya sea aumentando capacidad de procesamiento, memoria, etc. El problema principal es que es el coste de aumentar los recursos en un solo componente físico es exponencial y tiene un límite.

Escalado horizontal (scale out) es cuando se incrementa los componentes físicos o nodos en un sistema, si se incrementa la demanda de capacidad, se agregan más componentes físicos al sistema, sin tener que aumentar la capacidad de estos componentes.

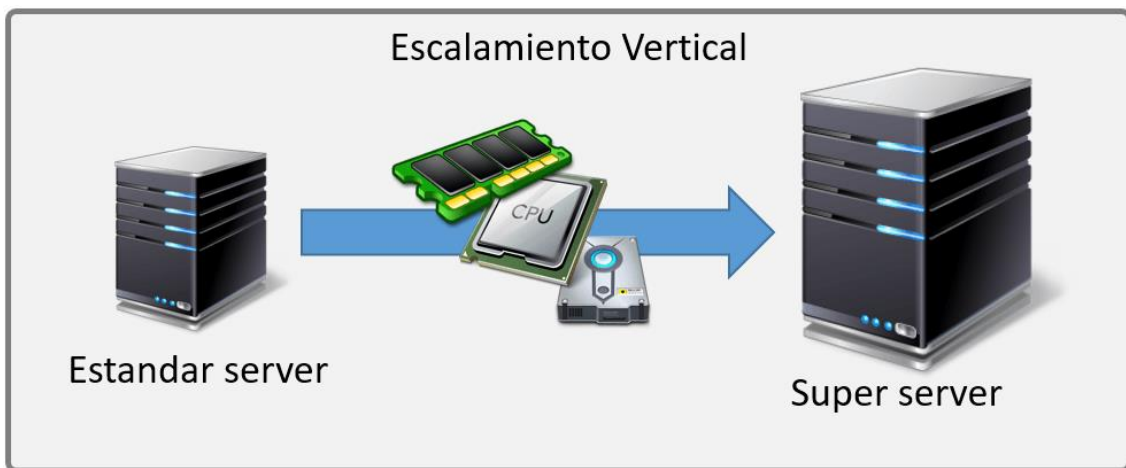


Figura 46-Escalamiento vertical (Fuente: Oscar Blancarte (2020). Introducción a la arquitectura de software- Un enfoque práctico, Primera Edición)

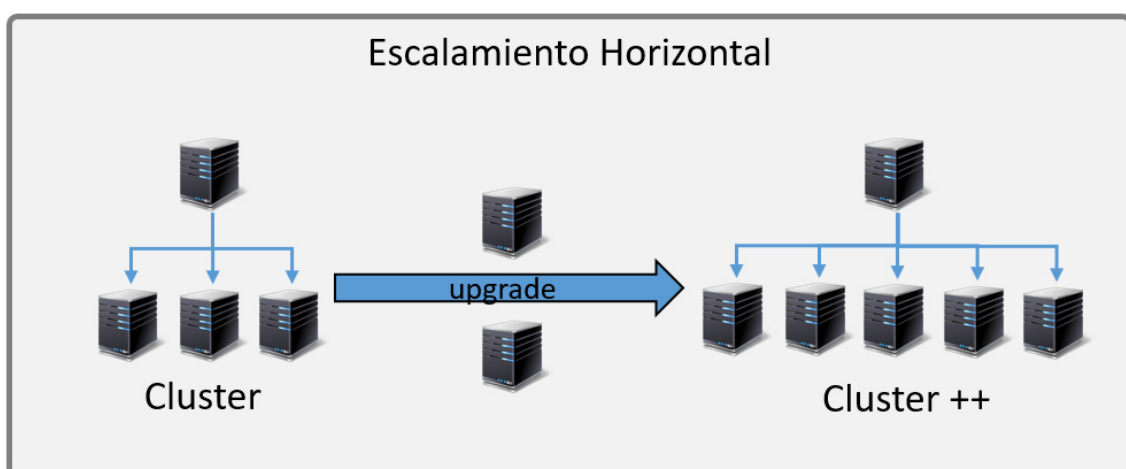


Figura 47-Escalamiento horizontal (Fuente: Oscar Blancarte (2020). Introducción a la arquitectura de software- Un enfoque práctico, Primera Edición)

Balance de carga

Es la técnica para dividir la carga de trabajo entre varios procesos, está muy relacionada con los sistemas que se despliegan sobre varios componentes físicos o nodos. El objetivo es evitar los cuellos de botellas a través de un algoritmo que balancea las peticiones al sistema entre los diferentes componentes físicos. Existen dos tipos:

A.- Balance de Carga por Hardware

- Realiza un balanceo de carga a través de un algoritmo y selecciona el nodo de menos carga de trabajo.
- La selección del nodo está basada en el algoritmo de sustitución y es aleatoria.
- La selección aleatoria puede provocar problemas en mantener la sesión del usuario por lo tanto puede condicionar el diseño de un sistema.
- El desarrollador mantiene la sesión.
- Es muy rápido, al ser procesos hardware.

B.- Balance de carga por Software

- Garantiza mantener la sesión de usuario ya que se examina a nivel de protocolo HTTP.
- Las peticiones del usuario son respondidas por el mismo nodo del sistema.
- Son más lentos que el balance por hardware.
- Son soluciones diseñadas en los componentes del software.

Arquitectura ESB

Es una solución de integración distribuida, basada en los eventos de mensajes. El objetivo de un bus de servicios empresariales (ESB) es proporcionar una comunicación segura entre los diferentes recursos tecnológicos (aplicaciones, componentes y servicios) distribuidos en la variedad de sistemas de la organización.

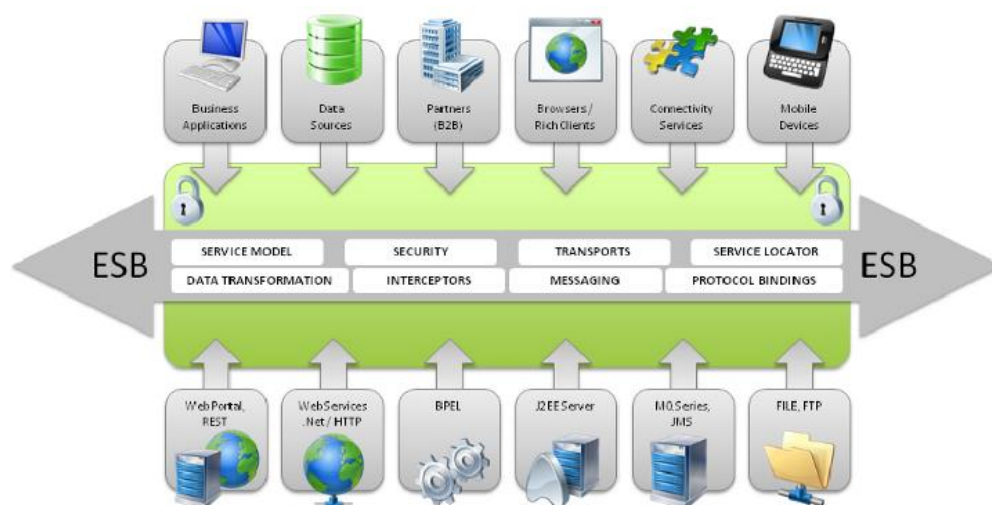


Figura 48-Diagrama de una arquitectura ESB (Fuente: Aziz Omer, Farooq Shoaib (2020). Research Trends in Enterprise Service Bus (ESB) Applications: A Systematic Mapping Study)

Requisitos de una arquitectura empresarial ESB.

Las siguientes características son esenciales para el diseño de una arquitectura empresarial ESB.

- Mensajería distribuida. Proporciona un método de transporte seguro y distribuido de mensajes respaldados por un mecanismo de almacenamiento y reenvío que garantiza la entrega en caso de problemas en la red.

- Transparencia de las ubicaciones. La mediación entre servicios permite que un cliente realice peticiones al proveedor sin la necesidad de saber la ubicación de su ejecución, de tal forma que si falla o se modifica su ubicación no es necesario notificar las actualizaciones a los clientes. Ayuda a minimizar los riesgos y costos de gestión.

- Transparencia del transporte. Proporciona protocolo de transporte físico para realizar la comunicación entre todos los servicios utilizando diferentes canales.

- Soporte multiprotocolo. Soporta muchos tipos de sistemas de transporte para la integración de diferentes componentes dispares en la arquitectura. Por ejemplo: JMS que posee más fiabilidad en el transporte asíncrono.

- Calidad de servicio. Hace referencia a su fiabilidad en la entrega de los mensajes y la fiabilidad del servicio de invocación, son puntos críticos para medir la calidad.

- Patrones de intercambio de mensajes. Hace referencia al soporte de dos paradigmas: La de solicitud/respuesta donde el cliente lanza la petición y espera la respuesta de manera síncrona. Y la de publicación/subscripción donde el cliente lanza la petición y se suscribe a la repuesta en vez de esperar a recibirla.

- Transformación. La flujos de actividades de un ESB permite dirigir mensajes de un servicio a otro por lo tanto debe ser capaz de transformar los datos de un formato a otro cuando el siguiente servicio lo necesita.

3.2.5.2 MARCO LEGAL

El conjunto de normas adoptadas con el objetivo de asegurar el cumplimiento de los procedimientos aduaneros se detallan en la tabla 41.

Tabla 41. Normas aduaneras (Fuente: SUNAT – Legislación Aduanera)

Código	Norma	Nombre	Publicación	Vigencia
GJA-01	Ley 27444	Ley del Procedimiento Administrativo General	11.04.2001	11.10.2001
GJA-02	Ley 27584	Ley que regula el Proceso Contencioso Administrativo	07.12.2001	16.04.2002
GJA-03	Decreto Legislativo 1053	Ley General de Aduanas	27.06.2008	28.06.2008(Entró en vigencia sólo el artículo 31º con excepción del inciso d)) 17.03.2009(Entró en vigencia parcialmente conforme a su Reglamento) 01.10.2010(Entró plenamente en vigencia)
GJA-04	DS.342-2016-EF	Arancel de Aduanas	16.12.2016	01.01.2017
GJA-05	Ley 28008	Ley de los Delitos Aduaneros	19.06.2003	28.08.2003

3.2.6 IMPLEMENTACIÓN DE LAS AREAS, PROCESOS, SISTEMAS Y SUS BUENAS PRÁCTICAS

La implementación de la arquitectura de integración se relaciona con los procesos de transmisión de las siguientes áreas ó unidades de negocios de TPP (Descritas en el alcance organizacional):

- A.-Naves.
- B.-Recaladas.
- C.-Contenedores.
- D.- Bill of Lading.

La organización de las unidades de negocios ha tenido que adaptar sus procesos para la atención de las transmisiones, para la cual cada unidad de negocio cuenta con los siguientes equipos de trabajo:

- A.- Analista de negocio, responsable de definir los documentos a transmitir a SUNAT.
- B.- Equipo técnico de operaciones, responsables de monitorear el estado de las transmisiones a SUNAT.
- C.- Supervisor desarrollo TI, responsable de gestionar la integración tecnología de su área con la arquitectura de transmisiones.

Seguidamente se han integrado a la arquitectura de transmisiones los siguientes sistemas:

- A.- Sistema Integral de Terminal de Almacenamiento (SITAC).
- B.- Servicios Navieros Portuarios (SINAV).
- C.- ERP SAP, Módulo de Compras, Logística.

Resultante de la experiencia se podrían considerar los siguientes pasos para la implementación de una nueva transmisión:

- El usuario solicita el envío de un nuevo documento de transmisión a SUNAT.
- El analista de negocio define el alcance funcional del nuevo documento (campos, información, plantilla).
- El supervisor de desarrollo TI solicita a la fábrica de software la configuración y el desarrollo del nuevo documento a transmitir.
- El supervisor de desarrollo TI coordina en su unidad de negocio los ajustes necesarios del sistema que enviará el nuevo documento a transmitir.
- Se realizan las pruebas de transmisión entre el sistema de la unidad de negocio, la arquitectura de integración y SUNAT.

3.3 EVALUACIÓN

3.3.1 EVALUACIÓN ECONÓMICA

La evaluación económica del costo del proyecto incluye lo siguiente, cabe precisar que los montos son aproximados:

Tabla 42. Costos del proyecto

COSTOS DEL PROYECTO				
Personal requerido	Costo por mes (S/.)	# de personas	# de meses	Costo total (S/.)
Product Owner	10,000.00	1	4	40,000.00
Scrum master	8,000.00	1	4	32,000.00
Analista de negocios	7,000.00	1	2	14,000.00

Analista de calidad	7,000.00	1	2	14,000.00
Arquitecto de software	9,000.00	1	4	36,000.00
Desarrollador	5,000.00	2	1	10,000.00
SUBTOTAL COSTO PERSONAL REQUERIDO (SCPR)				146,000.00
Recursos físicos en nube (Amazon Web Services)	Costo alquiler por mes (S/.)	# de instancias	# de meses	Costo total (S/.)
EC2 Large	352.00	1	3	1,056.00
EC2 Medium	102.00	2	3	612.00
Amazon RDS (db.t3.medium)	186.00	1	3	558.00
Amazon Kafka (kafka.m5.large)	544.00	1	2	1,088.00
SUBTOTAL COSTO RECURSOS FÍSICOS (SCRF)				3,314.00
COSTO TOTAL DEL PROYECTO (SCPR + SCRF)				149,314.00

3.3.2 INTERPRETACIÓN DEL VAN Y DEL TIR

La implementación del proyecto logró que durante los primeros 3 meses en producción la empresa tuvo un ahorro mensual de S/.20,000 por concepto de multas aplicadas por SUNAT.

Se estima que el primer año se ahorre S./ 100,000 , se proyecta el mismo ahorro durante los primeros 5 años a pesar del incremento de transacciones.

$$VAN = \sum \frac{FC_n}{(1 + TPD)^n}$$

En donde:

FC_n son los resultados de los flujos de caja para cada período

TPD es la tasa de descuento privada

n es el número de períodos del PIP

Para el flujo señalado:

Tabla 43. Cálculo del VAN

$= \frac{-149,314.00}{(1.13)^0} + \frac{100,000.00}{(1.13)^1} + \frac{100,000.00}{(1.13)^2} + \frac{100,000.00}{(1.13)^3} + \frac{100,000.00}{(1.13)^4}$
--

$$= -149,314.00 + 88,495.57 + 78,314.66 + 69,309.67 + 61,334.64$$

$$= 148,140.54$$

Si el VAN es mayor a 0 a una tasa de descuento privada de 13%, se puede concluir que el proyecto es rentable.

Se calcula el TIR con un VAN = 0, el resultado es igual a 56.00%.

Se considera rentable si el TIR es mayor al TPD.

A efectos del presente proyecto, el TIR (56.00%) es mayor que el TPD (13%).

Se puede concluir que el proyecto es rentable.

CAPITULO IV. REFLEXION CRITICA DE LA EXPERIENCIA

La experiencia obtenida del presente trabajo de suficiencia profesional se centra en los conocimientos técnicos sobre como mejorar la implementación de arquitecturas de integración que sean escalables.

Posterior a la implementación de la arquitectura en producción, se pudo obtener algunas métricas de los atributos de calidad del comportamiento de los componentes software:

- Performance: Los tiempos de respuesta que se tuvieron fueron adecuados según los incrementos transaccionales.
- Seguridad: Las transmisiones de documentos fueron encapsuladas en formato codificado para mayor seguridad. También se pudo restringir la comunicación a un rango de IPs previamente configurados. No se ha reportado ningún incidente de vulneración.
- Disponibilidad: La disponibilidad es al 100% ya que está desplegado en una plataforma en nube que asegura que el sistema responda las solicitudes en cualquier momento.
- Funcionabilidad: El sistema ha cumplido con el objetivo de transmitir documentos aduaneros en el tiempo máximo de control solicitado por SUNAT.

Esta información de métricas obtenidas permite mejorar los conocimientos arquitectónicos que cumplan los atributos de calidad que tienen un impacto directo sobre el diseño de arquitecturas de integración.

La aplicación de la nueva arquitectura permitió también que las unidades de negocios sean beneficiadas en sus actividades diarias trayendo los siguientes beneficios:

- Mejora de la cadena logística debido a que se garantiza la agilidad de los procesos de las unidades de negocios.
- Reducción de costos por multas debido al incumplimiento del control aduanero.
- Aumento en la confianza de los usuarios y operadores por la notable reducción de errores.
- Mejora del control mediante el monitoreo en línea de los estados e información de respuestas de los documentos aduaneros.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

La implementación de una arquitectura ESB mediante el integrador empresarial wso2 para la transmisión de documentos de terminales portuarios a SUNAT fue adecuadamente implementado solucionando los problemas de escalabilidad, y cumpliendo los atributos de calidad de diseño de software.

- Se analizó y definió el mapeo de los campos de negocios de los documentos aduaneros a transmitir.
- Se diseñó y especificó los componentes de la base de datos de Transmisiones.
- Se diseñó los diagramas y flujos de transmisión de los componentes de la arquitectura de integración.
- Se desarrolló los componentes de la arquitectura de integración usando los artefactos de la plataforma empresarial wso2.
- Se definió los escenarios de pruebas de estrés para la medición del rendimiento de escalabilidad.
- Se diseñó los componentes de despliegue de la arquitectura de integración.
- Se desarrolló los componentes backend y frontend del monitoreo de documentos transmitidos.

5.2 RECOMENDACIONES

Se recomienda una propuesta a futuro de mejora con machine learning basado en un análisis de información en tiempo real de los envíos y respuestas de las transmisiones a SUNAT que tendrían los siguientes objetivos:

- Predicción de errores realizando una matriz de escenarios de comportamientos.
- Alerta ante el acercamiento de umbrales que identifiquen un error.
- Recomendaciones sobre la gestión de recursos (memoria, CPU, Espacio, Seguridad) en nube.
- Identificación de componentes de software que hacen cuello de botella.

FUENTES DE INFORMACIÓN

Portal de SUNAT (2019). Operatividad Aduanera de Manifiesto de Ingreso. Obtenido de <https://www.sunat.gob.pe/operatividadaduanera/fast/estructuras/formato-Mani-Ingreso.html>

Portal de SUNAT (2019). Operatividad Aduanera de Manifiesto de Salida. Obtenido de <https://www.sunat.gob.pe/operatividadaduanera/fast/estructuras/formato-Mani-Salida.html>

Portal de SUNAT (2020). Manifiesto de Carga. Obtenido de <https://www.sunat.gob.pe/legislacion/procedim/despacho/manifiestos/procGeneral/despa-pg.09.htm>

Portal de Amazon Web Services (2020). Tipo de Instancias. Obtenido de <https://aws.amazon.com/es/ec2/instance-types/>

Ken Schwaber, Jeff Sutherland (2017). Guía de Scrum.

Satpathy, T. (2017). Guía para el Cuerpo de Conocimiento de Scrum, Tercera Edición.

Blancarte, O. (2020). Introducción a la arquitectura de software- Un enfoque práctico, Primera Edición.

Díaz, M. (2008). El papel de un bus de servicios empresariales (ESB) en una SOA.

Aziz Omer, Farooq Shoaib (2020). Research Trends in Enterprise Service Bus (ESB) Applications: A Systematic Mapping Study.

GLOSARIO

- ANDROID: Sistema operativo móvil de código libre.
- API: Interfaz de programación de aplicaciones, permite la comunicación entre dos componentes software.
- API GATEWAY: Permite unificar la publicación de APIs.
- ASTERISK: Programa de software libre que proporciona funcionalidades de una central telefónica.
- AWS: Amazon Web Services, plataforma para servicios en nube.
- BITBUCKET: Servicio de alojamiento web.
- CLOUD: Servicios en nube.
- CUSCAR- XML: Formato electrónico de intercambio de datos en XML.
- DOCKER: Tecnología de contenedores de código abierto que automatiza los despliegues de aplicaciones.
- ESB: Enterprise Service Bus, arquitectura que permite la integración de servicios y aplicaciones.
- GIT: Software de control de versiones.
- IRM: Ingreso y recepción de mercancías
- IOS: Sistema operativo móvil licenciado de Apple.
- JAVA: Lenguaje de programación para el desarrollo de software.
- ONPREMISE: Software que se instala y se ejecuta en los equipos de la organización.
- RSA. Registro de salida de mercancías.
- SEIDA: Sistema electrónico de intercambio de datos de SUNAT.
- SIX: Producto de core bancario.
- SPRING BOOT: Framework para el desarrollo de aplicaciones en Java.
- SUNAT: Superintendencia Nacional de Administración Tributaria.
- TPP: Terminales Portuarios Peruanos.
- WEBLOGIC: Servidor de aplicaciones.
- WSO2: Plataforma de integración empresarial que permite implementar arquitecturas de ESB.

ANEXOS

- 1. ANEXO 01 – Certificado de trabajo en Arquitectura Cloud**

CERTIFICADO DE TRABAJO

La Sra. ROSA MERCEDES PEÑA RIVERA, identificada con DNI N° 40799662, Representante Legal de la Empresa COMPLEXLESS SAC., con RUC 20543314529.


CERTIFICA:

Que, el Sr. VIDAL CUPE QUISPE, identificado con DNI N° 42575888, ha laborado en nuestra empresa como CONSULTOR EN ARQUITECTO CLOUD, bajo la modalidad de renta de cuarta categoría, desde el 26/10/2020 hasta el 30/04/2021, demostrando durante su permanencia responsabilidad, capacidad técnica y dedicación en las labores que le fueron encomendadas.

Se expide la presente a solicitud del interesado, para los fines que crea conveniente.

Lima, 30 de Abril del 2021

Atentamente,


ROSA M. PEÑA RIVERA
Representante legal
COMPLEXLESS SAC.

ROSA M. PEÑA RIVERA
Representante Legal
COMPLEXLESS SAC.

Escaneado con CamScanner

2. ANEXO 02 – Estándares de Base de Datos

TERMINALES PORTUARIOS PERUANOS

Estándares de Base de
Datos

Abril, 2013

Tabla de contenidos

1.1. Base de Datos	4
1.2. Nomenclatura	4
1.2.1. Grupos Lógicos en	4
1.2.2. Archivos Físicos de Datos	4
1.2.3. Archivo Físico de Transacciones	5
1.2.4. Tamaño Inicial de los Archivos Físicos de Datos y Transacción	5
1.2.5. Crecimiento de los Archivos Físicos de Datos y Transacción	5
1.2.6. Tamaño Máximo de los Archivos Físicos de Datos y Transacción	5
1.2.7. Otros Datos de Configuración	5
1.3. Diagramas	5
1.4. Tablas	6
1.4.1. Nombre	6
1.4.2. Tipos de Datos Definidos por el Usuario	7
1.4.3. Tipos de Datos Definidos para Parámetros de Valores de Tabla	7
1.4.4. Campos	8
1.4.5. Primary Key	10
1.4.6. Alternate Key	10
1.4.7. Inversion Entry Key	10
1.4.8. Foreign Key	10
1.4.9. Inversion Foreign Key	10
1.5. Consultas de Manipulación de Datos (DML)	11
1.6. Vistas	11
1.7. Procedimientos Almacenados	12
1.7.1. Información de Documentación	13
1.7.2. Sentencias Iniciales de Procedimientos Almacenados	13
1.7.3. Parámetros de Procedimientos Almacenados	13
1.7.4. Parámetros de Retorno y Retorno de Identity al Agregar	13
1.7.5. Control de Errores, retorno de Error o Éxito y Transacciones.	14
1.7.6. Recuperación Error o Éxito de un procedimiento Almacenado.	15
1.7.7. Uso de RAISERROR.	15
1.8. Funciones de Usuario	15
1.9. Variables	16
1.10. Cursores	16
1.11. Tablas Temporales	16
1.12. Variables Tablas	16
1.13. Permisos Usuarios	17
1.14. Funciones de Bases de Datos	17
1.15. Check Constraint – Reglas de Validación de Datos.	17
1.16. Valores Predeterminados (default)	17
1.17. Triggers	17
1.18. Campos Estándar de Uso en las Tablas	18

CONTROL DE CAMBIOS Y EVOLUCIÓN DEL DOCUMENTO				
Versión <99>	Fecha de Cambio <DD/MM/AA>	Modificado por <Persona que crea o modifica el documento>	Págs. <Número de página modificada>	Resumen del cambio <Describir el cambio, indicar sección o párrafo en la que realizó la modificación>
01	15/04/2013	Pedro Valle	Todas	Creación del Documento

1.1. Base de Datos

Se utilizará PostgreSQL como servidor de base de datos

1.2. Nomenclatura

Las base de datos se nombrarán utilizando el prefijo BD seguido de un guión bajo y luego el nombre de la base de datos, basándose en la Empresa, Grupo de Empresas o alguna particularidad que la identifique. De preferencia el nombre debe ser descriptivo pero podría utilizarse abreviaciones. Las bases de datos antiguas quedaran como están.

Ej.:

BDD_GEVERANDES	Base de Datos Grupo
BDD_GRUPOEVERANDES	Base de Datos Grupo
BDD_TPP	Base de Datos Grupo
BDD_GrupoTPPOLAP	Base Grupo TPP OLAP
BDD_GrupoTPPReporte	Base Grupo TPP Reporte

1.2.1. Grupos Lógicos en

Se utilizarán Grupos Lógicos para agrupar las tablas según su área de origen o de uso, cada grupo lógico tendrá archivos físicos separados, los grupos tendrán las iniciales GRP seguido de un guión bajo, 3 caracteres para la nomenclatura del área o departamento de la empresa

Ej.:

GRP_MGE	Tablas Generales
GRP_DOC	Tablas Documentación
GRP_OPE	Tablas Operaciones
GRP_SEG	Tablas Seguridad

Para los casos que se necesite tener grupos lógicos que agrupen tablas de un mismo documento, se agregará a lo anterior un guión bajo seguido del nombre que agrupa a las tablas o la abreviación utilizada para agrupar a las tablas.

GRP_MGE_Ente	Tablas Generales Ente
GRP_DOC_MDO	Tablas Manifiesto Documentario
GRP_OPE_COP	Tablas Control Operativo
GRP_SEG_Auditoria	Tablas Seguridad Auditoria

El Grupo Lógico PRIMARY que viene por defecto se utilizara lo menos posible.

1.2.2. Archivos Físicos de Datos

Será del mismo nombre de la base de datos, seguido del nombre del Grupo Lógico con el que está relacionado sin el prefijo GRP, agregándose un guión bajo y el año y mes de creación.

El Tamaño de cada archivo físico será en un inicio de 64mb y tendrá un incremento de 64mb hasta 2gb. Solo en casos especiales que se tenga que

aumentar el tamaño de incrementación deberá ser una progresión aritmética de 2ⁿ

1.2.3. Archivo Físico de Transacciones

Será del mismo nombre de la base de datos, seguido de guión bajo y la nomenclatura Log, la ubicación será en la misma carpeta que los archivos físicos de datos.

Ej.:

D:\DATA\BD_GEVERANDES\BD_GEverandes_Log
D:\DATA\BD_GEVERANDES\BD_GrupoEverandes_Log

En caso se necesitase agregar otros archivos físicos de transacción, se agregará después de Log un guion bajo y el año y mes de creación.

Ej.:

D:\DATA\BD_GEVERANDES\BD_GEVERANDES_Log_201201

1.2.4. Tamaño Inicial de los Archivos Físicos de Datos y Transacción

Tamaño inicial de 64mb.

1.2.5. Crecimiento de los Archivos Físicos de Datos y Transacción

El crecimiento y reserva de espacio será de 64mb, salvo casos en que se necesita mayor reserva por las operaciones realizadas, se aumentará el tamaño basándose siempre en una progresión aritmética de 2ⁿ

1.2.6. Tamaño Máximo de los Archivos Físicos de Datos y Transacción

El tamaño máximo de un archivo físico será 2gb.

1.2.7. Otros Datos de Configuración

El resto de datos de configuración de la base de datos se dejará tal como está por defecto, en caso se necesite cambios adicionales se incluirán en este documento.

1.3. Diagramas

Comenzará con las letras dgr, seguido de tres letras que identificará al sistema o subsistemas al que pertenece, seguido de un guión bajo y una palabra que identifique al diagrama claramente.

Ej.:

DGR_GEN_Ubicacion (Diagrama de las tablas necesarias para el mantenimiento de trabajadores)
DGR_OPE_Recepcion (Diagrama de las tablas necesarias para la recepción de mercancía)

1.4. Tablas

1.4.1. Nombre

Comenzará con el prefijo TB seguido de guión bajo, luego 3 letras que identificará que área se origina mayoritariamente los datos o esta mas involucrada principalmente, seguido de un guión bajo y el nombre que identifica a la tabla, el cual tendrá relación en lo posible con los datos que se almacenan. El nombre de la tabla será en singular.

Ej.:

TBL_OPE_Almacen	Logística, Tipos de Almacén
TBL_LOG_Almacen	Logística. Almacenes
TB_SEG_UsuarioGrupo	Seguridad Usuario Grupo
TB_SEG_AuditoriaAcceso	Auditoria de Accesos

Si la tabla tiene uso por varias áreas sin tener un área específica o que no pueda determinarse su mayor uso, se procederá a una nomenclatura general para el área con las siglas MGE (Maestro Genérico). Las tablas que se necesiten en áreas diferentes pero que tienen el mismo significado y almacena los mismos datos, también se utilizará MGE para la identificación del área.

Ej.:

TBL_MGE_Persona	Genérico de Persona
TBL_MGE_Pais	Genérico de País
TB_MGE_Ubigeo	Genérico de Ubigeo

Caso de Tablas que almacenan lo mismo y se crea como una sola tabla.

Ej.:

TB_OPE_Almacen y TB_LOG_Almacen se llamaría
TB_MGE_Almacen Logística. Almacenes

Las tablas que requieran agruparse para identificar un documento en su conjunto, después de la nomenclatura del área y el guión bajo, se utilizará una nomenclatura de 3 letras que representen a dicha agrupación de tablas para el documento, seguido del nombre de la tabla.

Ej.:

TB_OPE_ControlRecepcion	(Control Recepción Cabecera)
TB_OPE_ControlRecepcionDetalle	(Control Recepción Detalle)
TB_OPE_ControlRecepcionCaja	(Control Recepción Detalle Cajas)

1.4.2. Tipos de Datos Definidos por el Usuario

Se podrán utilizar Tipo de Datos definidos por el usuario, los cuales serán creados con la siguiente nomenclatura:

Deberán comenzar con TDDU seguido de un guión bajo y a continuación la descripción del tipo definido.

Ej.:

TDU_ID	int	identity, No Nulo
TDU_Activo	bit	Default 1 Validacion 0,1 No Nulo

```
CREATE TABLE TB_MGE_Tipo (
  cpnID_Tipo TDU_ID ,
  cpcTipo varchar(60) not null DEFAULT "",
  cpnEstado TDU_Activo
)
```

Se tendrán tipos definidos para los siguientes campos:

TDU_ID	Identificador del campo de la tabla.
TDU_FechaCreador	Fecha/Hora creación Registro
TDU_UsuarioCreador	Usuario Creador del Registro
TDU_EstacionCreador	Estación Creadora del Registro
TDU_FechaActualizador	Fecha Última Actualización
TDU_UsuarioActualizador	Usuario Última Actualización
TDU_EstacionActualizador	Estación Última Actualización

Las fechas de creación y actualización deberán ser asignadas por el servidor de bases de datos. En el INSERT INTO deben omitirse o comentarse, para el UPDATE la fecha de modificación debe asignarse con la fecha del servidor.

1.4.3. Tipos de Datos Definidos para Parámetros de Valores de Tabla

Deberán comenzar con TDDUTVP seguido de un guión bajo, a continuación, el nombre de la tabla que principalmente recibirá los datos. Si es necesario más de un parámetro para la misma tabla, se deberá agregar alguna descripción después del nombre de la tabla.

Ej.:

PVT_TB_MGE_Empresa	Para inserción masiva de Empresas.
PVT_TB_MGE_Persona	Para inserción masiva de Personas.
PVT_TB_MGE_Usuario	Para inserción masiva de Usuarios.

1.4.4. Campos

1.4.4.1. Campos No Claves

Comenzarán con las letras **cp**x y un nombre para identificar su atributo, donde **x** define el tipo de dato del campo.
NO deberán permitir NULL, salvo excepciones.
Deberán tener valores por defecto, salvo excepciones.

Los valores definidos para **x** son:

- n Tipos de datos numéricos, moneda, decimal, bit, que tengan que ver con números.
- c Tipo de datos varchar, char que tengan que ver con string.
- d Tipo de datos timestamp que tengan que con fechas.
- i Tipo de datos que tenga que ver con imágenes.
- o Tipo de datos que tenga que ver con objetos.

Ej.:

cpnCantidadDetalleCaja	Cantidad de Cajas del Detalle de Cajas
cpcDescripcionTipoAlmacen	Descripción Tipo de Almacén
cpdFechaNacimiento	Descripción Tipo de Almacén
cpITrabajador	imagen de Trabajador

1.4.4.2. Campos Claves Identity o Identificador de Registro

Comenzarán con las letras **cpnID** seguido de un guión bajo y el nombre de la tabla, sin el prefijo TBL y sin las 3 letras que identifican en que área se originan. Será de tipo identity y por lo tanto entero, para tablas de movimientos, se utilizara el tipo de dato de SQL Bigint. Estos campos no deberán permitir NULL

Ej.:

Tabla	Campo
TBL_LOG_TipoAlmacen	cpnID_TipoAlmacen
TBL_MGE_Pais	cpnID_Pais
TBL_LOG_Almacen	cpnID_Almacen
TBL_OPE_Recepcion	cpnID_Recepcion
TBL_FAC_VENVentaCab	cpnID_VENVentaCab
TBL_SDA_VENMercancia	cpnID_SDAMercancia

1.4.4.3. Permitir Null - Excepciones

Los campos no deberán permitir NULL, excepto:

Campo Fecha, si inicialmente al crearse el registro no es necesario este dato, según análisis.

Campos Foráneos o que vienen de otras tablas, siempre y cuando al crearse el registro inicialmente no se registra, según análisis.

Otros casos, deberá analizarse si es necesario o no.

1.4.4.4. Tabla Referencia para Asignar Null, Constraint y Valor Defecto.

Todos los campos char, varchar o sean string, no deben permitir NULL, y se debe asignar valor por defecto (""), siempre y cuando no se este obligado a registrar un dato, esto según análisis.

Los campos de tipo numérico y que son utilizados para realizar operaciones matemáticas (suma, resta, multiplicación, etc.), no deben permitir NULL.

Tipo	Null	Constraint	Defecto
Bigserial	NO	> 0	Correlativo
Campo foráneo, dato Obligatorio	NO	Dato de referencia	Si es necesario
Campo foráneo, dato NO obligatorio	SI	Dato de referencia	Si es necesario
String, dato Obligatorio	NO	<> ""	Si es necesario
String, dato NO obligatorio	NO	Si es necesario	""
Numérico, siempre contendrá un dato.	NO	Si es necesario	0 u otro.

Los campos de auditoria para la creación de un registro son obligatorios, Usuario Creador, Fecha Creación, PC Creación, deberán ser NO NULL, además de no permitir espacios. Los valores por defecto podrán ser:

Usuario Creador CURRENT_USER
 Fecha Creación CURRENT_TIMESTAMP()
 PC Creación HOST_NAME()

Los campos de auditoria para la modificación de un registro NO son obligatorios, Usuario Modificador, PC Modificación, deberán ser NO NULL, pero permitirán espacios, el campo de Fecha Modificación, permitirá NULL.

Las fechas de Creación y Modificación se grabarán del servidor de base de datos.

Otras situaciones se analizarán.

1.4.5. Primary Key

Comenzarán con las letras **PK** seguido de un guión bajo y el nombre de la tabla.

Ejm:

PK_TB_MGE_Trabajador
PK_TB_SEG_Usuario

También pueden generar automáticamente por un programa.

1.4.6. Alternate Key

Comenzará con las letras **AK**, seguido de un guión bajo y los nombres de los campos que se utilizan, puede utilizarse la que por defecto le asigna un diseñador de entidad relación.

Ejm:

AK_cpcApellidoPaternoopcApellidoMaterno
AK_ApellidoPaternoMaterno

1.4.7. Inversion Entry Key

Comenzará con las letras **IE**, seguido de un guión bajo y los nombres de los campos que se utilizan, puede utilizarse la que por defecto le asigna un diseñador de entidad relación.

Ejm:

IE_cpApellidoPaternoopcApellidoMaterno
IE_ApellidoPaternoMaterno

1.4.8. Foreign Key

Comenzarán con las letras **FK**, seguido de un guión bajo y los nombres de los campos foráneos, puede utilizarse la que por defecto le asigna un diseñador de entidad relación.

Ejm:

FK_SAT_VBFVendorBFCab_SAT_VBFVendorBFDet
FK_SAT_SLCSecueLlenaCab_SAT_SLCSecueLlenaDet
FK_PER_Trabajadores_SAT_ConstanDevolu
FK_PER_Trabajadores_SAT_ConstanDevolu_2

1.4.9. Inversion Foreign Key

Comenzará con las letras **IF**, seguido de un guión bajo, el nombre de la tabla hija y el nombre de la tabla padre, puede utilizarse la que por defecto le asigna un diseñador de entidad relación.

Ejm:

IF_SAT_VBFVendorBFDetSAT_VBFVendorBFCab
IF_SAT_SLCSecueLlenaDetSAT_SLCSecueLlenaCab
IF_SAT_ConstanDevoluPER_Trabajadores

1.5. Consultas de Manipulación de Datos (DML)

- 1.5.1. Utilizar TAB para indentación de código.
- 1.5.2. Evitar en lo posible el uso de * en el SELECT, escribir los nombres de los campos a recuperar.
- 1.5.3. Evitar en lo posible el uso de tablas temporales, utilizar en su reemplazo variables tipo tabla.
- 1.5.4. Evitar en lo posible el uso de cursores.
- 1.5.5. Cada nueva referencia a otra tabla, debe empezar en otra línea en el FROM comenzando con la relación entre las tablas.
- 1.5.6. Utilizar el nombre de la tabla como alias o quitando los prefijos, evitar el uso de alias de solo una letra. Si la misma tabla se utiliza varias veces en la misma consulta, escribir como alias el mismo nombre agregando algo que la diferencia o en todo caso un nombre que permita identificar su identidad claramente.

Ej.:

```
SELECT
  a.cpcNombEmpresa,
  b.cpcTipoDocumento
FROM TB_MGE_Empresa a INNER JOIN
  TB_MGE_Tipodocumento b ON A.cpnID_TipoDocumento = B.cpnID_TipoDocumento
Deberá ser:
```

```
SELECT
  TB_MGE_Empresa.cpcNombEmpresa,
  TB_MGE_Tipodocumento.cpcTipoDocumento
FROM TB_MGE_Empresa
  INNER JOIN TB_MGE_Tipodocumento ON TB_MGE_Empresa.cpnID_TipoDocumento =
  TB_MGE_Tipodocumento.cpnID_TipoDocumento
```

- 1.5.7. En las relaciones del FROM, escribir la relación en la columna siguiente y a continuación la tabla con su respectiva comparación, solo si se requiere utilizar AND para otra comparación de la relación, comenzar con al AND en una siguiente línea.
- 1.5.8. En la relación del JOIN el primer campo de la comparación será con referencia a la referenciada y el segundo campo con referencia a la tabla referencia.
- 1.5.9. Las palabras reservadas deben ir mayúsculas

1.6. Vistas

Comenzarán con las letras vis, seguido de la tabla principal a la que se hace referencia

Ej.:

```
visTB_SAT_Clientes (Subsistema Almacenes Textiles, Clientes)
visTB_MGE_Ubigeo (Subsistema General Ubigeo (Dist, Prov y Depar)
```

visTB_SAT_ClienteLogistico (Subsistema Almacenes textiles, Cliente Logístico)

1.7. Procedimientos Almacenados

- a. Comenzará con las letras pal, seguido del nombre de la tabla, y terminarán con identificaciones para indicar los procesos básicos de Agregar, Modificar, Eliminar, Obtener, Listar y Reportes.

Ej.:

palTB_MGE_EmbarcadorAgr	Agregar
palTB_MGE_EmbarcadorMod	Modificar
palTB_MGE_EmbarcadorEli	Eliminar
palTB_MGE_EmbarcadorObt	Obtener Registro, solo los de la tabla
palTB_MGE_EmbarcadorLst	Obtener Lista, asociados a otras tablas.
palTB_MGE_EmbarcadorRep	Reporte de Embarcadores
palTB_MGE_EmbarcadorCbo	Para llenar listas, combo box de clave valor
palTB_MGE_EmbarcadorGri	Para utilizar en grillas.

En base a esta nomenclatura y para procedimientos especiales, se le agregará algo más.

- a. Se tendrá un procedimiento especial para cambio de Estado de un registro que indica eliminación lógica.

palTB_MGE_EmbarcadorModEstado	Modificar Estado Registro
-------------------------------	---------------------------

- b. En base a esta nomenclatura y para procedimientos especiales, se le agregará después de Agr, Mod, Eli, etc, la descripción especial de lo que se quiere realizar con dicha opción.

palTB_MGE_EmbarcadorObtActivos	Obtener Registro de los activos.
palTB_MGE_EmbarcadorLstActivos	Obtener Lista con descripciones, asociados a otras tablas Activos.
palTB_MGE_EmbarcadorAgrSuplentes	Agregar especial.
palTB_MGE_EmbarcadorModEspecial	Modificar especial.

- c. Para procedimientos almacenados de inserciones masivas utilizando XML, se agregará después de Agr, Mod, Eli, etc, las letras XML y a continuación

palTB_MGE_EmbarcadorAgrXML	Agregar mediante XML
----------------------------	----------------------

- d. Para procedimientos almacenados de inserciones masivas utilizando Parametros de Valores de Tabla, se agregara despues de Agr, Mod, Eli, etc, las letras TVP y a continuación si es necesario una descripción.

palTB_MGE_EmbarcadorAgrPVT	Agregar mediante Parametros de Valores de Tabla
----------------------------	---

- e. Para otros procedimientos almacenados, utilizar un nombre que esté relacionado con la tabla de donde principalmente se obtienen los datos.

palTB_MGE_EmbarcadorLstPendiente	
----------------------------------	--

paITB_MGE_FACFacturaLstSinRecalada

1.7.1. Información de Documentación

Los procedimientos almacenados, tendrán una cabecera documentaria indicando los siguientes datos:

Autor-Creación
Descripción
Modificación

Este será el formato en el procedimiento almacenado.

```
-- *****  
-- Autor - Creación      : Generador - 07/04/2011  
-- Descripción          :  
--  
-- Modificaciones       : Controlado por TFS  
-- *****
```

1.7.2. Sentencias Iniciales de Procedimientos Almacenados

Los procedimientos almacenados, deberán incluir al inicio la desactivación del eco de las instrucciones ejecutadas, además también para los reportes o recuperación de datos exhaustiva la desactivación de la transacción.

1.7.3. Parámetros de Procedimientos Almacenados

Los nombres de parámetros recibidos cuando se trate de nombres de campos, será el mismo nombre del campo y seguirá las mismas reglas de creación de campos.

Ej.: cpnID_TipoDespacho
cpcTipoDespacho

Todos los parámetros deberán tener un valor por defecto en caso no reciban un valor.

```
Ej.: cpnID_TipoDespacho int = null OUTPUT,  
cpcTipoDespacho        varchar(50) = ''  
cpdFechaVigencia      datetime = CURRENT_TIMESTAMP  
cpcDocumento           char(15) = ''  
cpnI_Persona           int = null
```

Los parámetros que no tengan que ver con campos foráneos, seguir el mismo patrón.

1.7.4. Parámetros de Retorno y Retorno de Identity al Agregar

Los parámetros de retorno, tiene la misma nomenclatura que los del punto 1.6.1., el cual debe estar como OUTPUT.

Además, los procedimientos almacenados de Inserción, deben capturar el ID generado y ser retornado en el parámetro correspondiente.

1.7.5. Control de Errores, retorno de Error o Éxito y Transacciones.

Los procedimientos deberán devolver un valor de Error o Éxito al concluir. 0=Éxito,1=Error, esto servirá para cuando sea llamado por otro procedimiento almacenado reciba el resultado del procedimiento que se llamó. Además, se debe incluir una estructura de control de errores que permita bifurcar a una etiqueta de error y posterior retorno de ejecución con error. Las transacciones según sea necesario, se incluirá BEGIN TRANSACTION, ROLLBACK TRANSACTION y COMMIT TRANSACTION como parte del control de Errores.

```
BEGIN TRANSACTION
```

```
--Código.
```

```
IF ERROR <> 0 GOTO Error
```

```
Confirmar:
```

```
IF TRANCOUNT > 0  
    COMMIT TRANSACTION  
RETURN 0
```

```
Error:
```

```
IF TRANCOUNT > 0 ROLLBACK  
    TRANSACTION  
RETURN 1
```

El BEGIN TRANSACTION en caso se utilice, debe posponerse lo más que se pueda y utilizarse justo antes de INSERT, UPDATE o DELETE.

El código en rojo, debe ser incluido en todos los procedimientos almacenados, quitar los comentarios en caso se utilice transacciones.

1.7.6. Recuperación Error o Éxito de un procedimiento Almacenado.

Desde un procedimiento almacenado que llame a otro procedimiento almacenado, se deberá declarar una variable de tipo bit con el siguiente nombre `return_status` al inicio del procedimiento y asignar el valor 0, en esta variable se recibirá el resultado del procedimiento que fue llamado y luego pasado al tag de Error.

```
IF TRANCOUNT > 0
    COMMIT TRANSACTION
RETURN 0
```

Error:

```
IF TRANCOUNT > 0 ROLLBACK
    TRANSACTION
RETURN 1
```

1.7.7. Uso de RAISERROR.

El uso de RAISERROR, deberá utilizarse de la siguiente manera:

```
IF saldo < 0
    ERROR('No existe Saldo', 16, 1)IF
```

```
ERROR <> 0 GOTO Error
```

Esto permitirá devolver un mensaje con un número de error 50000 para que pueda ser controlado en los programas clientes. La siguiente línea permitirá enviar el control al tag de Error y así el procedimiento devolver 1.

1.8. Funciones de Usuario

Comenzará con las letras `fun`, seguido en lo posible del nombre de la tabla a la que principalmente se está haciendo referencia y un descriptivo de su uso. En todo caso si la función es para cuestiones genéricas, siempre debe comenzar con `funMGE_` seguido del guión bajo o con `fun` seguido de la nomenclatura del proceso o sistema que se está desarrollando.

Ej.:

funMGE_Feriado	Valor Escalar
funMGE_UltimoPeriodo	Valor Escalar
funMGE_Empresa	Valor Tabla

Se incluirá información de documentación del procedimiento almacenado al igual que el punto 1.7.1 de los procedimientos almacenados.

1.9. Variables

Comenzaran con @ seguido de lx, donde "l" representa variable local, x representa el tipo de datos según como se tiene para los campos de la tabla. Para variables de las tablas, se crearan como están en el campo de la tabla.

Ej.:

cpnID_Almacen	Variable del campo
InDato	Variable de tipo numérica.
lcDato	Variable de tipo string.

1.10. Cursores

Los cursores comenzaran con las letras cur, seguido del nombre de la tabla, tabla temporal o variable tabla que se quiera recorrer.

Ej.:

curMGE_Empresa Para recorrer la tabla MGE_Empresa
 curMGE_FACFacturaCab Para recorrer la tabla MGE_FACFacturaCab

1.11. Tablas Temporales

Tratar de no utilizar, salvo mucha necesidad, en vez de estas utilizar las variables tabla, que utilizan memoria en vez de disco. Tendrán la siguiente nomenclatura empezar con # seguido de tem y un guión bajo, para reconocer que es una tabla temporal, seguido preferentemente con el nombre de tabla que originan los datos de mayor predominio.

Ej.:

#temMGE_Empresa Tabla temporal de MGE_Empresa
 #temSIU_ComprobantePagoCab Tabla temporal de SIU_ComprobantePagoCab

1.12. Variables Tablas

Utilizan la memoria y son más eficientes que las tablas temporales, utilizar estas en vez de las tablas temporales. Las variables tablas tendrán la siguiente composición, empezaran con @ seguido de vta, para reconocer que es una variable tabla, seguido preferentemente con el nombre de tabla que originan los datos de mayor predominio.

Ej.:

vtaMGE_Empresa Variable tabla de MGE_Empresa
vtaSIU_ComprobantePagoCab Variable tabla de SIU_ComprobantePagoCab

1.13. Permisos Usuarios

Se crearán nombres de usuarios y se aplicará la seguridad según las políticas establecidas.

1.14. Roles de Bases de Datos

Comenzará con las letras rbd, seguido del nombre de la función, los usuarios y permisos serán asignados según las políticas de seguridad establecidas.

Ej.: rbd_TPP (Funciones especiales a usuarios de TPP)

1.15. Check Constraint – Reglas de Validación de Datos.

Comenzarán con las letras **CHK** seguido de un guión bajo y la descripción. No se usarán reglas mediante sp_bindrules.

Ej.:

CHK_MayorCero El valor debe ser mayor a cero
CHK_NoEspacios No permitir espacios.

Todos los campos deberán tener validaciones de datos mínimos siempre y cuando no sean campos identity o con alguna excepción.

Así tenemos:

Tipo
char, varchar o cualquier campo de tipo cadena
No permitir Espacios, salvo excepciones.

integer, smallint o cualquier campo de tipo numérico
No permitir Negativos, salvo excepciones.

1.16. Valores Predeterminados (default)

Empezarán con DF seguido de un guión bajo y la descripción, estos serán asignados en los campos de las tablas.

Ej.:

DEF_Cero 0
DEF_Espacio ""

1.17. Triggers

Comenzarán con las iniciales tri, seguido del nombre de la tabla, y al final Agr, para indicar que es al agregar, Mod al modificar y Eli al Eliminar. Dependiendo de la necesidad de otros triggers, se agregará además descripción para el fin creado.

Ej.:

triTB_MGE_EmbarcadorAgr	Al Agregar
triTB_MGE_EmbarcadorAgrEspecial	Al Agregar Especial
triTB_MGE_EmbarcadorEli	Al Eliminar
triTB_MGE_EmbarcadorMod	Al Modificar

1.18. Campos Estándar de Uso en las Tablas

Se utilizarán campos en las tablas para identificar lo siguiente y que representaran lo mismo en todas las tablas que se utilizan.

1.18.1. Campos de Control y Auditoría

Fecha/Hora Creador	cpdFechaCreador
Usuario Creador	cpcUsuarioCreador
Estación Creador	cpcEstacionCreador
Fecha/Hora Modificador	cpdFechaModificador
Usuario Modificador	cpcUsuarioModificador
Estación Modificador	cpcEstacionModificador

De uso en todas las tablas que se crean.

1.18.2. Campos de Estado de Registro

Estado Registro	cpnID_EstadoRegistro
Estado Eliminado	cpnEliminado
Estado Duplicacion	cpnDuplicacion

Estado Registro: Sera un campo proveniente de otra tabla donde tendrá una lista de Estados de Registro, los códigos predeterminados a utilizar serán:

1	Activo
2	Inactivo
3	Anulado

Activo e Inactivo, de uso más frecuente en Catálogos o maestros.

Activo y Anulado, de uso más frecuentes en documentos o transaccionales.

