



**UNIVERSIDAD CENTRAL DE VENEZUELA**  
**FACULTAD DE CIENCIAS**  
**POSTGRADO EN CIENCIAS DE LA COMPUTACIÓN**

***ARQUITECTURA ORIENTADA A SERVICIOS PARA LA  
CONCILIACIÓN DE LA TESORERÍA INTERNACIONAL***

**Lic. Jonathan José Valera Carmona**  
**Tutora: Dra. Vanessa Leguizamo**

Caracas - Venezuela  
Junio, 2018



## VEREDICTO

Quienes suscriben, miembros del jurado designado por el Consejo de la Facultad de Ciencias de la Universidad Central de Venezuela, para examinar el **Trabajo de Grado** presentado por: **JONATHAN JOSÉ VALERA CARMONA** Cédula de identidad N° 19.671.811, bajo el título "ARQUITECTURA ORIENTADA A SERVICIOS PARA LA CONCILIACIÓN DE LA TESORERÍA INTERNACIONAL", a fin de cumplir con el requisito legal para optar al grado académico de **MAGÍSTER SCIENTIARUM, MENCIÓN CIENCIAS DE LA COMPUTACIÓN**, dejan constancia de lo siguiente:

1. Leído como fue dicho trabajo por cada uno de los miembros del jurado, se fijó el día 07 de Febrero de 2019 a las 09:00 AM., para que el autor lo defendiera en forma pública, lo que éste hizo en el Auditorio Manuel Bemporad, mediante un resumen oral de su contenido, luego de lo cual respondió satisfactoriamente a las preguntas que le fueron formuladas por el jurado, todo ello conforme con lo dispuesto en el Reglamento de Estudios de Postgrado.
2. Finalizada la defensa del trabajo, el jurado decidió **aprobarlo**, por considerar, sin hacerse solidario con la ideas expuestas por el autor, que se ajusta a lo dispuesto y exigido en el Reglamento de Estudios de Postgrado

Para dar este veredicto, el jurado estimó que el trabajo examinado constituye un aporte para el desarrollo de soluciones de software basadas en Arquitecturas Orientadas a Servicios, en este caso para el proceso de negocio de conciliación de tesorería internacional.

En fe de lo cual se levanta la presente ACTA, a los 07 días del mes de Febrero del año 2019. Conforme a lo dispuesto en el Reglamento de Estudios de Postgrado actuó como Coordinadora del jurado la Dra. Ana Vanessa Leguizamo.

Dr. Andrés Sanoja / C.I. 11.229.248  
UCV

Dra. Claudia León / C.I. 8.969.546  
BGV-UCV

Dra. Ana Vanessa Leguizamo / C.I. 13.309.245  
UCV  
Tutora



POSTGRADO EN CIENCIAS  
DE LA COMPUTACION  
Universidad Central de Venezuela

UNIVERSIDAD CENTRAL DE VENEZUELA

Facultad de Ciencias

Maestría en Ciencias de la computación

**“ARQUITECTURA ORIENTADA A SERVICIOS PARA LA CONCILIACIÓN DE LA TESORERÍA INTERNACIONAL”**

Autor: Lic. Jonathan J. Valera. C

Tutora: Dra. Vanessa Leguizamó

**Resumen**

En el presente estudio se planteó como objetivo; desarrollar un sistema bajo una Arquitectura Orientada a Servicio (SOA) implementando *Representational State Transfer* (REST). Utilizando la metodología *Service Oriented Modeling and Architecture* (SOMA) que permite identificar, especificar y realizar servicios, garantizando la calidad de los mismos. La propuesta surgió cuando observamos que el Departamento de Inversiones de la Tesorería Internacional (DITI) de la Gerencia de Administración de Reservas Internacionales (GARI), del Banco Central de Venezuela (BCV), carecía de la automatización de algunos de sus procesos. En esa instancia se gestiona los activos y/o recursos de dicha Institución, concretamente el *Cash* y oro. La Tesorería Internacional cuenta con cuatro (4) procesos de negocios dentro del DITI, a saber: **1-La conciliación;** que es el proceso donde parte su negocio, en el cual inician sus actividades. **2-La carga en sistemas;** corresponde a la carga de los saldos conciliados en el Sistema Automatizado de Tesorería Internacional (SATI), adicionalmente cargan en el sistema de gestión de cartera PORTIA los datos de inversiones automáticas, las tasas de interés, intereses generados, entre otros, **3- La generación de reportes de los balances de los saldos;** tales como saldos por corresponsal, cartas de créditos, los flujos de fondos, entre otros y, **4-El proceso de revisión de *Intradays*;** que es la revisión de las transacciones que se han ejecutado durante el día para compararlas con la estimación prevista. En esta investigación se empleó el método científico que se inició a través de la observación, complementándolo con un diagnóstico de la información previa existente en el DITI. Por tal situación, haciendo uso de algunas herramientas tecnológicas como JavaEE y WildFly Application Server, con el firme propósito que se facilite la conciliación de la data suministrada por cada corresponsal a través de los mensajes *The Society for Worldwide Interbank Financial Telecommunications* (SWIFT). Ello, pues con esa mensajería se logra que el Instituto desarrolle los procesos de conciliación mas eficientemente, y la conciliación se suministre a los demás sistemas que existen actualmente en el BCV. Concluyéndose que las pruebas sistemáticas desarrolladas de los servicios realizados con la herramienta SoapUI mejora la calidad de los servicios expuestos que soportan el proceso de negocio de la conciliación, facilitando la integración con otros sistemas del Banco.

**Palabras claves:** Arquitectura Orientada a Servicios (SOA), conciliación bancaria, Banco Central de Venezuela (BCV), tesorería internacional, SOMA, REST

## Índice general

Resumen .....	2
Índice de tablas.....	6
Índice de figuras.....	9
Introducción.....	10
<i>CAPITULO I</i> .....	12
Planteamiento del problema .....	12
Importancia y justificación de la investigación .....	14
Antecedentes .....	15
Propuesta de Investigación .....	16
Objetivo general.....	16
Objetivos específicos.....	16
<i>CAPITULO II</i> .....	17
Marco Teórico Referencial .....	17
Departamento de Inversiones y Tesorería Internacional (DITI).....	17
Conciliación bancaria .....	18
Portafolio de inversión.....	18
Proceso de negocio .....	19
Servicios Web .....	19
Arquitectura Orientada a Servicios .....	19
Manifiesto SOA.....	20
SOAP y REST .....	20
REST como solución a implementar.....	21
Orquestación y Coreografía de servicios.....	22
Orquestación como solución a implementar.....	23
Metodología.....	23
<i>CAPITULO III</i> .....	27
Marco Aplicativo .....	27
Fase 1. Modelado del negocio y transformación .....	27
Especificación de Requisitos .....	29
Proceso de Conciliación de la Tesorería Internacional del BCV .....	29
Introducción .....	29
Propósito .....	30

Ámbito del Sistema .....	30
Definiciones, Acrónimos y Abreviaturas .....	31
Descripción general .....	31
Perspectiva del producto .....	31
Funciones del producto.....	31
Características de los usuarios .....	34
Restricciones .....	34
Suposiciones y dependencias.....	34
Requisitos específicos .....	35
Atributos del sistema .....	38
Fase 2. Identificación.....	39
Casos de uso.....	39
Modelación de casos de uso alineados a SOA .....	39
Capa de proceso del consumidor .....	41
Capa de proceso de larga duración.....	45
Capa de proceso de corta duración .....	50
Modelo de servicio derivado del Caso de Uso .....	56
Modelado de servicios de objetivos.....	57
Aplicar pruebas decisivas de servicio .....	61
Garantizar que el servicio está alineado con la empresa.....	61
Garantizar que el servicio es componible .....	61
Garantizar que el servicio es reutilizable .....	63
Garantizar que el servicio es técnicamente viable.....	63
Modelado de servicios: Capa de abstracción lógica de servicios.....	63
Modelo de servicios .....	64
Identificar objetivos y subobjetivos .....	65
Fase 3. Especificación de los servicios.....	67
Arquitectura Funcional.....	67
Vista de comportamiento .....	86
Modelado de componentes de servicios .....	92
Contrato de servicios.....	95
Requisitos no funcionales de los servicios .....	103
Fase 4. Realización de servicio .....	103

Asignación de los servicios a los componentes.....	105
Decisiones arquitectónicas relacionadas con la realización del servicio .....	105
CAPITULO IV.....	106
Implementación y despliegue .....	106
Fase 5. Implementación .....	106
Estándares para la codificación en Java .....	107
Paquetes de código fuente .....	107
Clases e interfaces.....	107
Herramienta para la implementación del proceso de negocio.....	108
Implementación del proceso de negocio Conciliar Operaciones.....	108
Desarrollo de la aplicación web asociados a la interfaz de usuario del proceso .....	110
Descripción de la aplicación web .....	110
Especificación de los módulos.....	112
Fase 6. Despliegue y pruebas .....	114
Plan de pruebas unitarias.....	115
Plan de pruebas de componentes.....	115
Plan de pruebas de sistema .....	116
Plan de pruebas de usuario .....	116
Cómo se va a probar .....	116
CAPITULO V.....	125
Conclusiones .....	125
Referencias bibliográficas .....	127

## Índice de tablas

Tabla 1 Descripción del proceso de conciliación de la Tesorería Internacional.....	28
Tabla 2 Requerimientos del sistema .....	35
Tabla 3Requerimientos del sistema .....	35
Tabla 4Requerimientos del sistema .....	35
Tabla 5Requerimientos del sistema .....	36
Tabla 6Requerimientos del sistema .....	36
Tabla 7Requerimientos del sistema .....	36
Tabla 8 Requerimientos del sistema .....	37
Tabla 9 Requerimientos del sistema .....	37
Tabla 10 Requerimientos del sistema .....	37
Tabla 11 Requerimientos del sistema .....	38
Tabla 12 Requerimientos del sistema .....	38
Tabla 13 Requerimientos del sistema .....	38
Tabla 14 Servicios candidatos .....	59
Tabla 15 Prueba decisiva de servicios. Alineación de los servicios con la empresa .....	61
Tabla 16 Prueba decisiva de servicios. Componibilidad de los servicios .....	62
Tabla 17 Pruebas decisivas de servicio. Descripción de servicio externalizada .....	62
Tabla 18 Pruebas decisivas de servicios. Reutilización de servicios.....	63
Tabla 19 Matriz Procesos-Actividades-Objetivos.....	65
Tabla 20 Objetivos priorizados.....	66
Tabla 21 Identificar y asociar servicios con objetivos .....	66
Tabla 22 Eventos por operación de servicio MsgSwift .....	68
Tabla 23 Estructura de mensaje de entrada S01P01 .....	68
Tabla 24 Estructura de mensaje de salida S01P01.....	69
Tabla 25 Estructura de excepción de S01P01 .....	69
Tabla 26 Estructura de mensaje de entrada S01P02 .....	69
Tabla 27 Estructura de mensaje de salida S01P02.....	69
Tabla 28 Estructura de excepción de S01P02 .....	69
Tabla 29 Estructura de mensaje de entrada S01P03 .....	70
Tabla 30 Estructura de mensaje de salida S01P03.....	70
Tabla 31 Estructura de excepción de S01P03 .....	70
Tabla 32 Estructura de mensaje de entrada S01P04 .....	70
Tabla 33 Estructura de mensaje de salida S01P04.....	70
Tabla 34 Estructura de excepción de S01P04 .....	70
Tabla 35 Estructura de mensaje de salida S01P05.....	71
Tabla 36 Estructura de excepción de S01P05 .....	71
Tabla 37 Eventos por operación de servicio Catalogo .....	71
Tabla 38 Estructura de mensaje de entrada S02P01 .....	72
Tabla 39 Estructura de mensaje de salida S02P01.....	72
Tabla 40 Estructura de excepción de S02P01 .....	72
Tabla 41 Estructura de mensaje de entrada S02P02 .....	72

Tabla 42 Estructura de mensaje de salida S02P02.....	72
Tabla 43 Estructura de excepción de S02P02 .....	73
Tabla 44 Estructura de mensaje de entrada S02P03 .....	73
Tabla 45 Estructura de mensaje de salida S02P03.....	73
Tabla 46 Estructura de excepción de S02P03 .....	73
Tabla 47 Estructura de mensaje de salida S02P04.....	73
Tabla 48 Estructura de mensaje de excepción de S02P04.....	73
Tabla 49 Estructura de mensaje de entrada S02P05 .....	74
Tabla 50 Estructura de mensaje de salida S02P05.....	74
Tabla 51 Estructura de mensaje de excepción de S02P05.....	74
Tabla 52 Eventos por operación de servicio Instrumento .....	74
Tabla 53 Estructura de mensaje de entrada S03P01 .....	75
Tabla 54 Estructura de mensaje de salida S03P01.....	75
Tabla 55 Estructura de excepción de S03P01 .....	75
Tabla 56 Estructura de mensaje de entrada S03P02 .....	76
Tabla 57 Estructura de mensaje de salida S03P02.....	76
Tabla 58 Estructura de excepción de S03P02 .....	76
Tabla 59 Estructura de mensaje de entrada S03P03 .....	76
Tabla 60 Estructura de mensaje de salida S03P03.....	76
Tabla 61 Estructura de excepción de S03P03 .....	76
Tabla 62 Estructura de mensaje de entrada S03P04 .....	77
Tabla 63 Estructura de mensaje de salida S03P04.....	77
Tabla 64 Estructura de excepción de S03P04 .....	77
Tabla 65 Estructura de mensaje de entrada S03P05 .....	77
Tabla 66 Estructura de mensaje de salida S03P05.....	77
Tabla 67 Estructura de excepción de S03P05 .....	78
Tabla 68 Estructura de mensaje de salida S03P06.....	78
Tabla 69 Estructura de excepción de S03P06 .....	78
Tabla 70 Estructura de mensaje de entrada S03P07 .....	78
Tabla 71 Estructura de mensaje de salida S03P07.....	78
Tabla 72 Estructura de excepción de S03P07 .....	78
Tabla 73 Estructura de mensaje de entrada S03P08 .....	79
Tabla 74 Estructura de mensaje de salida S03P08.....	79
Tabla 75 Estructura de excepción de S03P08 .....	79
Tabla 76 Estructura de mensaje de entrada S03P09 .....	79
Tabla 77 Estructura de mensaje de salida S03P09.....	79
Tabla 78 Estructura de excepción de S03P09 .....	80
Tabla 79 Eventos por operación de servicio Solicitud.....	80
Tabla 80 Eventos por operación de servicio Solicitud.....	81
Tabla 81 Estructura de mensaje de entrada S13P01 .....	81
Tabla 82 Estructura de mensaje de salida S13P01.....	81
Tabla 83 Estructura de excepción de S13P01 .....	81
Tabla 84 Estructura de mensaje de entrada S13P02 .....	81
Tabla 85 Estructura de mensaje de salida S13P02.....	81
Tabla 86 Estructura de excepción de S13P02 .....	82



Tabla 87 Estructura de mensaje de salida S13P03.....	82
Tabla 88 Estructura de excepción de S13P03 .....	82
Tabla 89 Estructura de mensaje de salida S13P04.....	82
Tabla 90 Estructura de excepción de S13P04 .....	82
Tabla 91 Estructura de mensaje de salida S13P05.....	83
Tabla 92 Estructura de excepción de S13P05 .....	83
Tabla 93 Estructura de mensaje de entrada S13P06 .....	83
Tabla 94 Estructura de mensaje de salida S13P06.....	83
Tabla 95 Estructura de excepción de S13P06 .....	84
Tabla 96 Estructura de mensaje de entrada S13P07 .....	84
Tabla 97 Estructura de mensaje de salida S13P07.....	84
Tabla 98 Estructura de excepción de S13P07 .....	84
Tabla 99 Estructura de mensaje de entrada S13P08 .....	84
Tabla 100 Estructura de mensaje de salida S13P08.....	85
Tabla 101 Estructura de excepción de S13P08 .....	85
Tabla 102 Estructura de mensaje de entrada S13P09 .....	85
Tabla 103 Estructura de mensaje de salida S13P09.....	85
Tabla 104 Estructura de excepción de S13P09 .....	85
Tabla 105 Eventos por servicios. Catálogo.....	95
Tabla 106 Eventos por servicios. Catálogo.....	96
Tabla 107 Eventos por servicios. Intrumento .....	97
Tabla 108 Eventos por servicios. Instrumento.....	98
Tabla 109 Eventos por servicios. MsgSwift .....	99
Tabla 110 Eventos por servicios. Solicitud .....	101
Tabla 111 Decisiones de realización de servicio .....	104
Tabla 112 Asignación de los servicios a los componentes.....	105
Tabla 113 Decisiones arquitectónicas relacionadas con los servicios .....	105
Tabla 114 Decisiones arquitectónicas relacionadas con los servicios .....	105
Tabla 115 Formato de descripción SWIFT MT950 .....	108
Tabla 116 Módulos de la aplicación web .....	111
Tabla 117 Casos de pruebas con JUnit.....	118
Tabla 118 Casos de prueba con SoapUI .....	119

# Índice de figuras

Ilustración 1 Proceso de negocios para la conciliación en el DITI.....	13
Ilustración 2 Organigrama parcial del BCV. Vicepresidencia de Operaciones Internacionales. (Tomando del BCV, 2016).....	17
Ilustración 3 Estructura del catálogo de instrumentos.....	18
Ilustración 4. Ciclo de vida de SOMA (Arsanjani, 2004). ....	24
Ilustración 5. Pasos generales de SOMA (Arsanjani, 2004).....	24
Ilustración 6 Propuesta de estructuración de la cartera.....	32
Ilustración 7 Casos de Uso. Proceso de conciliar operaciones.....	40
Ilustración 8 Casos de Uso. Proceso de conciliar operaciones.....	41
Ilustración 9 Modelo de servicio derivado de los Caso de Uso.....	56
Ilustración 10 Capa de servicios.....	64
Ilustración 11 Arquitectura funcional SOA-Sistema de conciliación.....	67
Ilustración 12 Interfaz MsgSwift.....	68
Ilustración 13 Interfaz de servicio Catalogo.....	71
Ilustración 14 Interfaz del servicio de instrumento.....	74
Ilustración 15 Interfaz del servicio de Solicitud.....	80
Ilustración 16 Clases por servicios. Catálogo.....	86
Ilustración 17 Secuencia por servicios. Catálogo.....	87
Ilustración 18 Clases por servicios. MsgSwift.....	88
Ilustración 19 Clases por servicios. Instrumento.....	89
Ilustración 20 Clases por servicios. Solicitud.....	90
Ilustración 21 Secuencia por servicio. Solicitud.....	91
Ilustración 22 Componente de servicio. Catalogo.....	92
Ilustración 23 Componente de servicio. Instrumento.....	93
Ilustración 24 Componente de servicio. Solicitud.....	94
Ilustración 25 Vista del despliegue SOA en la plataforma.....	114
Ilustración 26 Inicio de sesión.....	121
Ilustración 27 Creación de un corresponsal.....	121
Ilustración 28 Catálogo de instrumentos.....	122
Ilustración 29 Solicitud de cobertura.....	123
Ilustración 30 Reporte: Montos no identificados.....	124

## Introducción

El presente trabajo se desarrolló con el propósito de ofrecer una propuesta de una arquitectura orientada a servicios para la conciliación de la Tesorería Internacional del Banco Central de Venezuela (en lo sucesivo, BCV). En este sentido, es importante significar que la tesorería es una de las disciplinas fundamentales dentro de la gestión financiera de las organizaciones (López y Soriano, 2014). Ella se centra en la gestión de las disponibilidades de la liquidez financiera. Algunos de los elementos con que trabaja la tesorería son el dinero en caja, los saldos disponibles y previstos en las diferentes cuentas bancarias, las inversiones financieras, los préstamos, y el control de los gastos e ingresos financieros. Su propósito es, por lo tanto, realizar una buena gestión del dinero y de sus respectivos movimientos. Se trata entonces de una optimización en la gestión de los recursos de liquidez financiera de la organización.

Nuestro objeto de estudio está circunscrito en el BCV, específicamente en el Departamento de Inversiones y Tesorería Internacional (DITI). El antes mencionado departamento tiene como propósito:

*“Diseñar e instrumentar las estrategias de inversión de los portafolios de reservas internacionales (divisas / oro monetario) y otros recursos en moneda extranjera cuya gestión haya sido confiada por ley al BCV, a los fines de procurar resultados eficientes conforme a las políticas, lineamientos y estrategias de inversión y riesgos vigentes. Asimismo, administrar el inventario de oro monetario y los flujos (divisas / oro monetario), con el objeto de controlar las tenencias de dichos recursos y dar cumplimiento a los compromisos contraídos” (BCV, 2012).*

Entre sus funciones establecidas se encuentra:

*“Centralizar, procesar y controlar los flujos en divisas y oro monetario manejados por el Instituto, con el objeto de proporcionar la cobertura necesaria para atender los compromisos contraídos y determinar el monto neto disponible para ser invertido.” (BCV, 2012).*

Como parte de la revisión de los procesos realizados por quienes integran el DITI, se busca obtener una oportuna respuesta en la gestión en esa área y detectar posibles debilidades, para aplicar los correctivos respectivos y necesarios para fortalecer y optimizar las prácticas diarias, obteniendo mayor eficiencia. Adicionalmente, se logró identificar otras debilidades, por ejemplo en el proceso de conciliación bancaria, donde recae una mayor repercusión.

Entre las labores que lleva a cabo el DITI, en el BCV, se encuentra el determinar diariamente la información real de liquidez. Es decir, al inicio de la jornada diaria

se debe saber con cuánto dinero cuenta la Institución, movimientos, entradas y salidas monetarias. Igual operación se debe contar al finalizar la jornada diaria. En la actualidad, el referido proceso, se desarrolla de manera manual y el o la analista, invierten un tiempo no menor a 70 minutos (más de una hora) para lograr su objetivo.

Por lo antes descrito, teniendo ubicado nuestro objeto de estudio, pasaremos a identificarlo y describirlo. En este orden de ideas, podemos definir nuestro objeto de estudio con una definición sencilla, de lo que significa gestión de liquidez (cash management), que es el conjunto de medidas estratégicas y organizativas que afectan a los flujos monetarios, en definitiva, los resultados financieros del BCV. Es decir, el DITI es responsable de la gestión de la liquidez del BCV. Abarcando desde la planificación de la inversión, la financiación, las relaciones bancarias hasta la gestión de la cobertura de saldo, entre otros. (Arellano, 2015).

Conscientes de la responsabilidad que cumple el DITI y la respuesta que deben emitir rápida y de manera eficiente, se plantea hacer una propuesta que agilice los procesos en esa instancia. Por ello, es importante contar con la automatización de procesos para una exitosa gestión en la DITI. Con la automatización de los procesos desarrollados manualmente, se pretende agilizar e incrementar la eficiencia y eficacia de los procesos ahí realizados.

Para el desarrollo del presente trabajo, se ha considerado implementar un sistema de conciliación bancaria basado en la utilización de tecnologías libres, permitiendo así tener interoperabilidad con otros sistemas del BCV. Para ello, se utilizó una Arquitectura Orientada a Servicios, basada en *Representational State Transfer* (REST), bajo un lenguaje de programación JAVA, implementación de Enterprise JavaBeans que forman parte del estándar de construcción de aplicaciones empresariales (JEE) de Oracle *Corporation*.

El presente trabajo de maestría se estructuró en capítulos, a saber:

- Capítulo I: El Planteamiento del Problema, donde se relata la situación actual del Departamento de Inversiones y Tesorería Internacional con el proceso de conciliación.
- Capítulo II: Marco Teórico Referencial, donde se describe el dominio del negocio, las tecnologías y herramientas utilizadas para el desarrollo de este trabajo de maestría.
- Capítulo III: Marco Aplicativo, se describe cada fase del método que se usó para lograr el objetivo propuesto en este trabajo.
- Capítulo IV: Implementación y despliegue, se continúa con las fases del método pero a nivel de implementación..
- Capítulo V: Conclusiones, se describe las experiencias de la realización de este trabajo de maestría y finalmente se añaden las referencias bibliográficas que se tomaron como apoyo para desarrollar el presente aporte.

# **CAPITULO I**

## **Planteamiento del problema**

En la actualidad el DITI del Banco Central de Venezuela (BCV), cuenta con un Sistema Automatizado de Tesorería Internacional (SATI) que está desarrollado bajo la herramienta PowerBuilder, de entorno gráfico de programación de clase empresarial elaborada por la empresa SyBase (Novalys, 2016). Sin embargo, SATI es un sistema que no cubre las expectativas de todos los requerimientos del DITI. Tal aseveración se realiza luego de realizar un diagnóstico en el área de DITI, que permitió identificar que algunos de los procesos que desempeña el DITI, para la fecha de iniciar el presente trabajo y aun estando ya culminado, en su mayoría son manuales. Sumado a ello, algunas operaciones que maneja el SATI, particularmente el control de los flujos de fondos no lo gestionan de manera óptima, por lo que se vuelve insuficiente para realizar todos los procesos de negocios del departamento. El referido sistema, le permite a los analistas, adscritos al departamento, un acceso limitado y restringido a la información, lo que implica el incremento de las deficiencias y el escaso control sobre aspectos tan importantes como son la conciliación bancaria, la posibilidad de contrastar fechas, el cálculo de intereses de cuentas bancarias o la previsión, control de gastos de transacciones, así como tampoco la posibilidad de generar reportes estadísticos de flujos de fondos de manera eficaz e inmediata.

Por consiguiente, los analistas del DITI se apoyan en el uso de hojas de cálculo para realizar el proceso de conciliación, que en función a la fecha valor, de las operaciones, pueden alcanzar entre 24 horas o 72 horas para hacerse efectivas. Eso significa que los profesionales del área, de manera manual invierten tiempo y esfuerzo, alcanzando un tiempo que fluctúa alrededor de 70 minutos diarios, para generar la conciliación. Ello origina un consumo innecesario de tiempo y esfuerzo para la entrega de la documentación correspondiente a las conciliaciones bancarias, efectuadas ante los departamentos pertinentes que emplean dicha información para la oportuna toma de decisiones, en cuanto al uso de los fondos económicos se refiere.

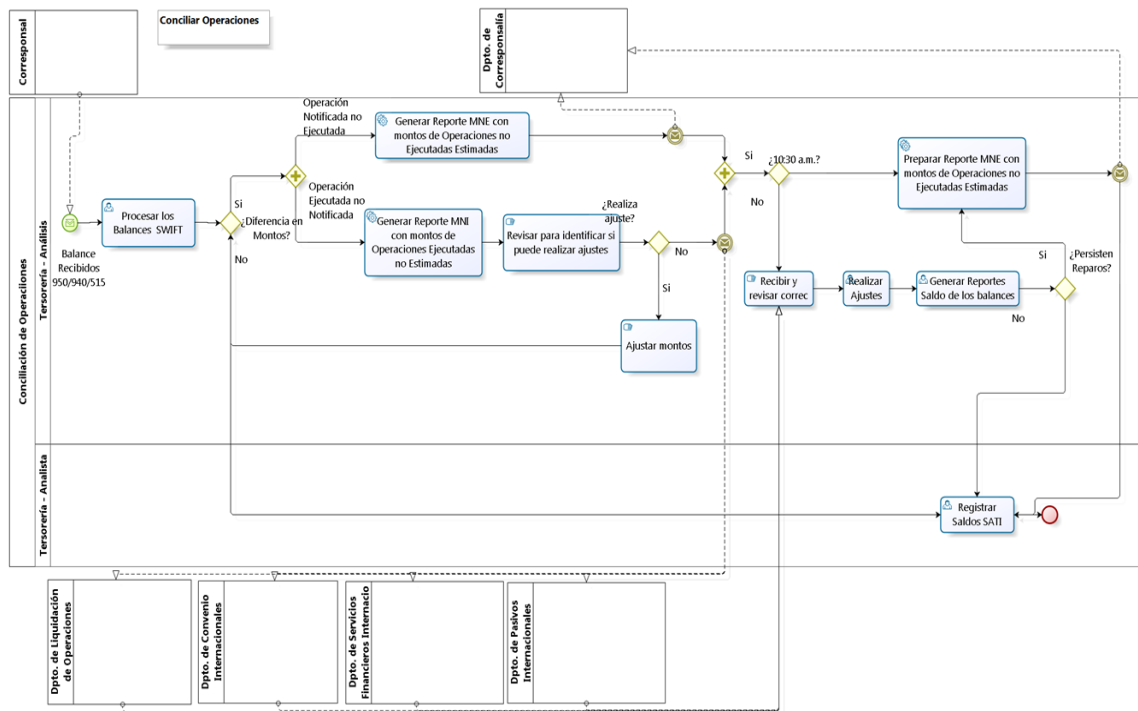
El manual de normas y procedimientos del DITI establece los pasos que debe realizar el Departamento para efectuar la conciliación de los movimientos de las cuentas en moneda extranjera, efectuada con los bancos corresponsales.

Por lo arriba descrito, y observando el problema que presenta el DITI, el sistema propuesto resuelve los procesos antes referidos, permitiendo:

- Controlar el cruce de las operaciones reflejadas en los balances de los bancos corresponsales contra los cortes de cuentas de la DITI.

- Determinar las partidas pendientes de atención y realizar la inclusión de las mismas para el respectivo cuadro de los bancos corresponsales.
- Informar a todas las unidades del BCV, que tengan partidas pendientes, a fin de que efectúen, de ser necesario, los registros contables requeridos para realizar los ajustes que se deriven de las conciliaciones efectuadas, o que realicen el seguimiento con el banco corresponsal correspondiente.
- Efectuar mensualmente el seguimiento de las cuentas de instrumentos, cuentas de gastos, intereses por saldo y estimaciones de ingresos y egresos.

En la Figura 1 se muestra el proceso de negocio del DITI a través de la notación de modelado de procesos de negocio (BPMN, por sus siglas en inglés), para en función a ello plantear una solución. (ver tabla 1 para la descripción)



bizagi

Ilustración 1 Proceso de negocios para la conciliación en el DITI

## Importancia y justificación de la investigación

La conciliación constituye un mecanismo de control de cualquier institución bancaria, en el caso que nos ocupa el BCV, que permite sustentar las diferencias entre el balance del corresponsal y la estimación hecha por el DITI, permitiendo verificar que se esté realizando una adecuada anotación de los cobros y pagos efectuados durante un determinado lapso, así como los gastos vinculados. La conciliación bancaria es un proceso desarrollado en el DITI del BCV, que permite comparar los balances emitidos por los corresponsales contra los estimados por esa instancia. Por tal situación, son competencias del DITI realizar las siguientes funciones:

- El reporte de montos no identificados y montos no ejecutados,
- los saldos de los balances, montos reales y estimados, y
- los cuadros de los balances

Todo ello, para determinar si existen diferencias entre los registros de la tesorería y los registros realizados por el corresponsal, con el firme propósito de asegurarse que todas las operaciones bancarias queden correctamente contabilizadas y reflejadas en los estados financieros del BCV. Es importante resaltar que en la actualidad el número estimado de documentos, que los analistas revisan y procesan aproximadamente, por día, alcanzan valores tan pocos como 40 balances desarrollados de manera manual, tal como habíamos referido con antelación en la introducción del presente aporte, es decir, es la cantidad en promedio a conciliar diariamente.

La presente investigación propuso un sistema basado en una Arquitectura Orientada a Servicios, que permitirá capturar la información base para generar las posiciones diarias y las provisiones (conciliación), entre los que se resaltan las siguientes:

- Recepción automática de balances a través del SWIFT.
- Conciliación bancaria.
- Manejo y seguimiento de las condiciones de los corresponsales.
- Seguimiento de los montos no identificados y montos no ejecutados.
- Generación de reportes del proceso de conciliación.

Al generar automáticamente los datos esenciales para la toma de decisiones y medición de resultados, con este sistema se le facilitará a la tesorería explicar las diferencias que pudiesen existir entre las cifras registradas del DITI y el estado de cuenta del corresponsal, además, determinar el importe real de efectivo en el corresponsal en una fecha y tiempo determinado. Por otra parte, el hecho de estar basado en una Arquitectura Orientada a Servicios, permite tener interoperabilidad con otros sistemas del BCV.

## Antecedentes

Son pocas las contribuciones que han atendido la perentoria necesidad de automatizar el departamento del DITI. No obstante, podemos nombrar que el antecedente más importante para la presente investigación es el Sistema Automatizado de Tesorería Internacional (SATI) del BCV. El SATI es la aplicación que actualmente usan en el DITI. Ello para llevar el control de los ingresos y egresos de las divisas con la que cuenta el Banco (reservas internacionales). A través de ella se reciben las solicitudes de coberturas de las unidades autorizadas del BCV, así como también las operaciones propias de la tesorería (inversiones, transferencias entre corresponsales, compra o venta de divisas, traspasos de instrumentos financieros, etc). El sistema no contempla el proceso de conciliar las operaciones, ni propias del DITI, ni ajenas, no genera reportes de la posición de los saldos en los corresponsales ni la capacidad de recibir y procesar los mensajes SWIFT que emiten los corresponsales por lo que esas tareas son hechas por los analistas financieros del DITI. La aplicación está hecha como sistema monolítico y no es capaz de interactuar con otros sistemas del Banco. Es constantemente mantenida y ajustada (en lo posible) según las necesidades del DITI, por los analistas de sistemas de la Gerencia de Sistemas e Informática del BCV.

Los sistemas informáticos tradicionales se han organizado en grandes bloques monolíticos que contienen tanto los procesos de negocio como sus funciones automatizados. El surgimiento del paradigma de Computación Orientado a Servicios para el desarrollo de aplicaciones, hoy día es una de las principales soluciones tecnológicas para todos esos procesos de negocios que requieren un sistema distribuido. El paradigma más popular para el desarrollo de software escalable e interoperable actualmente es del estilo arquitectónico orientado a servicios, aunque no es un concepto nuevo, data alrededor del año 2000 y facilita la integración e interacción con otros sistemas tanto propios como ajenos.

Entre los estudios realizados en relación a la conciliación bancaria podemos nombrar a: Martínez Cruz, O. et. al. (2014), quienes se plantearon como objetivo la creación de un sistema de conciliación y depuración bancaria. Utilizando una arquitectura Cliente-Servidor de 3 capas y una base de datos transaccional. Ellos utilizan como insumo para la entrada a su proceso unos archivos .TXT para el proceso de conciliación; demostrando que un sistema de conciliación bancaria resulta efectivo para la mejora de la gestión de tesorería.

Ahora bien, recientemente ubicamos que la compañía Grifols (2016), realizó un estudio de la conectividad y seguridad de SWIFT indicando la incidencia de implementar una solución de conciliación a través del producto SWIFT for Corporates. La compañía trabajaba en un sistema de gestión descentralizado, es decir, los departamentos de tesorería eran independientes por lo que se dificultaba



el manejo de saldos globales de la empresa. Su investigación permitió a la empresa centralizar la gestión a través de una única plataforma multibancaria con los mismos protocolos y el mismo nivel de seguridad en las distintas áreas de tesorería y gestión de efectivo.

Si bien esos estudios previos muestran la experiencia de implementar un sistema de conciliación, no deja de ser menos cierto que ofrecen ejemplos de la efectividad de utilizar un buen sistema de conciliación de saldos.

## **Propuesta de Investigación**

Este trabajo de investigación propone desarrollar un sistema de conciliación de las transacciones financieras operadas por el Departamento de Tesorería Internacional del BCV, con el objetivo de determinar el flujo de divisas, y así brindar una mayor eficiencia en el referido departamento, basado en una Arquitectura Orientada a Servicios, que permita agilizar el proceso de conciliación bancaria y a su vez interconectarse con otros sistemas del BCV para el intercambio de información. Para lograr la propuesta, nos formulamos las siguientes interrogantes:

¿Cómo abarcar eficientemente, desde una perspectiva arquitectónica de servicios, el proceso de conciliación y la integración con los sistemas heterogéneos del Banco Central de Venezuela? ¿Cuáles servicios hay que crear? Y en función a ello, ¿Cómo orquestar los servicios de manera que pueda proveer una conciliación de las operaciones financieras del Departamento de Tesorería Internacional del BCV?

Formuladas estas interrogantes y para obtener respuestas a nuestros planteamientos, nos propusimos alcanzar los siguientes objetivos:

### **Objetivo general**

Desarrollar un sistema de conciliación bancaria basado en los estándares de la Arquitectura Orientada a Servicios que permita determinar el flujo de divisas de la Tesorería Internacional del BCV

### **Objetivos específicos**

- Diagnosticar y analizar el proceso de conciliación bancaria que ejecuta el DITI
- Proponer una arquitectura orientada a servicios basada en REST, que permita interoperar con los demás sistemas del BCV.
- Identificar y desarrollar los servicios que permitan realizar la conciliación bancaria internacional y establecer su orquestación.
- Construir una aplicación que permita llevar a cabo la conciliación bancaria.
- Realizar pruebas de validación al sistema.
- Realizar pruebas de aceptación.

# CAPITULO II

## Marco Teórico Referencial

### Departamento de Inversiones y Tesorería Internacional (DITI)

El DITI adscrito a la Gerencia de Administración de Reservas Internacionales (ver figura 2) es el encargado de administrar todo el portafolio (oro monetario/divisas) de las reservas internacionales del BCV, a los fines de controlar las tenencias de esos recursos y dar cumplimiento a los compromisos contraídos.

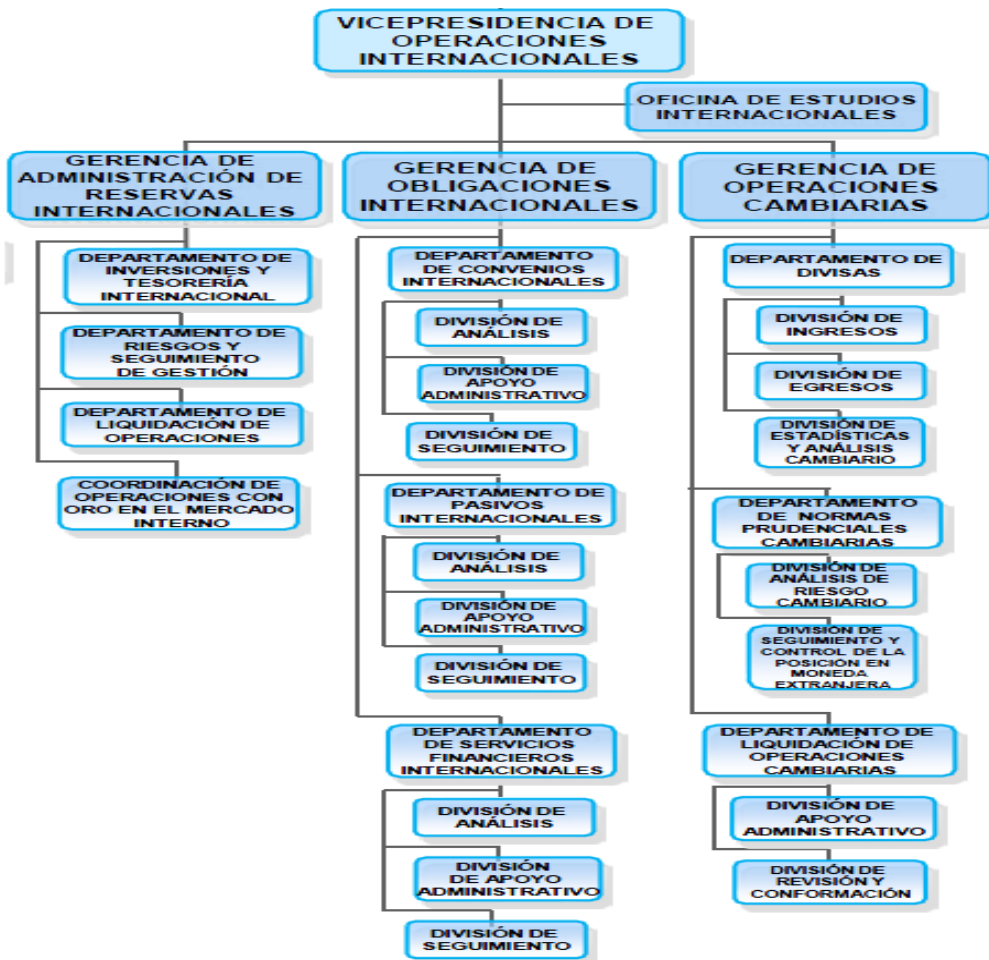


Ilustración 2 Organigrama parcial del BCV. Vicepresidencia de Operaciones Internacionales. (Tomando del BCV, 2016)

## Conciliación bancaria

La conciliación bancaria viene dada por la comparación, revisión, y respectivos ajustes entre las solicitudes de ingreso y/o egreso de fondos generadas por los departamentos de operaciones internacionales (ver figura 2) con los saldos de las cuentas corrientes de los corresponsales bancarios, a través de los balances o estados de cuentas de un período de tiempo específico y determinado. La revisión se realiza al comparar los saldos antes referidos en función de su fecha valor; en caso de existir diferencias se coteja las entradas y salidas notificadas por los departamentos, con los débitos y créditos registrados en las cuentas corrientes; finalmente, de haber diferencias, se procede a los ajustes necesarios.

## Portafolio de inversión

Uno de los elementos de la tesorería internacional es la organización de los portafolios de inversión. Al analizar un portafolio involucramos directamente las diferentes alternativas de inversión que satisfacen las necesidades de un inversionista y que maximizan su inversión. Un portafolio es el que determina la distribución óptima de sus inversiones, teniendo en cuenta la rentabilidad esperada y el riesgo de las diferentes alternativas de decisión (Cervantes, M. & Vergara, J., 2012)

En el contexto de este trabajo de grado de maestría, el proceso de conciliación requirió la organización y categorización de los instrumentos en los que se distribuyen los fondos con los que cuenta el BCV. Para esa categorización se creó un catálogo de instrumentos con la siguiente estructura:

- RIO (Reservas Internacionales Operativas) / OAME (Otros Activos en Moneda Extranjera)
  - ✓ PROPIETARIO - ORIGEN-FUENTE (Portafolio – Cartera)
    - TRAMO
      - ❖ Tipo instrumento
      - ❖ Banco
      - ❖ cuenta corriente
      - ❖ Plaza
      - ❖ Moneda
      - ❖ Mercado – Tipo de Cambio
      - ❖ Cuenta contable
      - ❖ Operación de negocio
      - ❖ Monto
      - ❖ Fecha

*Ilustración 3 Estructura del catálogo de instrumentos*

## **Proceso de negocio**

Los procesos de negocio involucran la captura de una secuencia ordenada de las actividades e información de apoyo. Un proceso representa lo que una organización realiza para lograr cumplir su propósito u objetivo. (Stephen, A. y Miers, D., 2009)

## **Servicios Web**

Somerville refiere el servicio web como

“una instancia de una noción más general de servicio, que se define un acto o una función ofrecidos por una parte a otra. Aunque el proceso puede asociarse a un proceso físico, la función es esencialmente intangible...” (Somerville, Ian. 2011).

Por consiguiente, los servicios web son sistemas software diseñados para soportar una interacción interoperable máquina a máquina sobre una red que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

## **Arquitectura Orientada a Servicios**

La Arquitectura Orientada a Servicios (SOA, Service Oriented Architecture) busca la alineación de los elementos de la Tecnología de la Información (TI) con los objetivos del negocio por medio de la organización general de dichos elementos, de manera que sean expuestos como una red de recursos integrados, simplificados y flexibles. SOA es un enfoque de diseño de arquitectura cuyo objetivo es integrar las distintas funcionalidades del negocio y exponerlas en forma de servicios. Es decir, son una forma de desarrollar sistemas distribuidos en la que los componentes del sistema son servicios independientes.

Es por eso, que Dunnavant, T y Johnston, G. (2011), en representación de IBM, definen a SOA como:

“Una forma de organizar y comprender organizaciones, comunidades y sistemas para maximizar la agilidad, la escala y la interoperabilidad. El enfoque de SOA es simple: las personas, las organizaciones y los sistemas brindan servicios a cada uno. SOA, entonces, es un paradigma arquitectónico para definir cómo las personas, organizaciones y sistemas proporcionan y usan servicios para lograr resultados” Ídem.

Es por ello que para la resolución del problema propuesto se hace uso de una SOA por las distintas aplicaciones en distintas tecnologías que existen en el BCV, que se requieren interactuar con el sistema de conciliación de la tesorería internacional facilitando la automatización y difusión de los servicios del proceso de negocio.

## **Manifiesto SOA**

Earl, T. (2009) ha establecido unos principios rectores al momento de implementar una arquitectura SOA. Entre esos principios se pueden rescatar los siguientes:

- Establecer un conjunto uniforme de estándares empresariales y de políticas basado en estándares de la industria, de facto, y de la comunidad

Los estándares de la industria representan especificaciones tecnológicas que ayudan a establecer, entre otras cosas, características básicas consistentes de la arquitectura de la tecnología. Sin embargo, el uso de estándares de la industria solamente, no garantiza que los servicios sean intrínsecamente interoperables.

Es así, que se establecieron criterios y se reforzaron los estándares de diseño en el departamento de la Coordinación Funcional de Operaciones Internacionales (CFOI) para garantizar la interoperabilidad de los sistemas.

- Verificar que los servicios satisfacen los requerimientos y las metas del negocio

Esto quiere decir que los servicios pueden ser mal utilizados. Al crecer y gestionar un portafolio de servicios, su uso y eficacia en el cumplimiento de los requerimientos de negocio deben ser verificados y medidos. Los servicios no deben ser utilizados sólo porque están disponibles, sino para verificar que ellos realmente están cumpliendo y satisfaciendo las necesidades y requerimientos de negocio.

Por ello se aplicó pruebas decisivas de servicio para la verificación y validación y corroborar que los servicios satisfacen su especificación y validar que también satisface las necesidades del DITI.

## **SOAP y REST**

Para la implementación de una arquitectura orientada a servicios con el uso de Web Services existen diversos estándares como Simple Object Access Protocol (SOAP) y Representational State Transfer (REST).

SOAP es un estándar de intercambio de mensajes que soporta la comunicación entre servicios descentralizado - distribuido. Es un protocolo basado en XML que consiste en definir un marco para describir cuál es el mensaje y cómo procesarlo (W3C, 2000).

Por otro lado, tenemos el estilo REST. Éste, es un estilo arquitectónico basado en la transferencia de representaciones de recursos de un servidor a un cliente. Es el estilo que subyace en la Web como un todo y se ha usado como un método mucho más simple que SOAP para implementar servicios Web. (Sommerville, Ian, 2011)

El estilo REST intenta emular al protocolo HTTP o protocolos similares mediante la restricción de establecer la interfaz a un conjunto conocido de operaciones estándar (por ejemplo GET, PUT, POST, DELETE). Por tanto, este estilo se centra más en interactuar con recursos que con mensajes y operaciones (Rafael Navarro, 2006).

## **REST como solución a implementar**

Para la solución de la propuesta de investigación se hace uso del estilo arquitectónico REST, principalmente, para cumplir con directrices organizacionales y debido a que gran parte de los sistemas que van a interactuar con la solución propuesta, son hechos en Power Builder y no son capaces de soportar SOAP, es por ello que la arquitectura orientada a servicios es con base al estilo arquitectónico REST. Para el uso de REST se hace uso de tecnologías y herramientas que lo soportan y que a continuación se mencionan:

- **Servidor de aplicaciones WildFly**

El servidor de aplicaciones WildFly, antiguamente conocido como JBoss, es un ambiente de ejecución de Red Hat para aplicaciones de Java que provee la infraestructura para ejecutar aplicaciones que manejan el negocio. Al estar basado en Java, WildFly puede ser utilizado en cualquier sistema operativo para el que esté disponible la máquina virtual de Java. Un servidor de aplicaciones sirve como una plataforma para desarrollar y construir los servicios web y los Enterprise Java Beans (EJB) ofreciendo una plataforma de alto rendimiento para aplicaciones de e-business combinando una arquitectura orientada a servicios SOA.

- **Java EE con JAX-RS y Jersey**

JAX-RS: Java API for RESTful Web Services es una API del lenguaje de programación Java que proporciona soporte en la creación de servicios web de acuerdo con el estilo arquitectónico Representational State Transfer (REST). JAX-

RS usa anotaciones, introducidas en Java SE 5, para simplificar el desarrollo y despliegue de los clientes y puntos finales de los servicios web.

Jersey es la implementación de referencia de la JAX-RS 2.0 (especificación JSR 339) Jersey implementa soporte para las anotaciones definidas en la JSR-311, lo que facilita a los desarrolladores crear servicios web RESTful con Java y la JVM de Java (Sunil Gulabani, 2013)

- **Spring Security**

Spring Security es un marco que se enfoca en proporcionar autenticación y autorización para aplicaciones Java. Brinda servicios integrales de seguridad para aplicaciones de software empresarial basadas en Java EE.

Las dos áreas principales de seguridad de las aplicaciones son "autenticación" y "autorización" (o "control de acceso"). Estas son las dos áreas principales que tiene como objetivo Spring Security

Spring Security es compatible con una amplia gama de modelos de autenticación. Aprovechando esa ventaja se implementa para la integración con LDAP a nivel organizacional.

- **Oracle Database**

El uso del gestor de base de datos de Oracle es meramente por tema organizacional. No obstante, se aprovecha el potencial de ese gestor de base de datos para implementar tipos de información que son de naturaleza jerárquica y están representados más convenientemente en una estructura de árbol. Esto, haciendo referencia al catálogo de instrumentos se buscó cómo representar dicha información jerárquica en una tabla relacional.

## **Orquestación y Coreografía de servicios**

La composición de servicios es la automatización de un proceso de negocio, de agrupar lógica de servicios funcionales entre los servicios de tarea, servicios de entidad y servicios de utilidad. Entonces, el componer servicios significa establecer mecanismos que permitan a dos o más de ellos cooperar entre sí para resolver requisitos que van más allá del alcance de sus capacidades individuales. De ahí que la composición representa la implementación de servicios web complejos que internamente invocan a otros servicios web (Zulma Díaz, 2006).

La orquestación es componer servicios y el orden en que son ejecutados para alcanzar el objetivo de un proceso de negocio, es decir, es la integración de varios servicios web con la finalidad de originar nuevos procesos o servicios de mayor nivel para dar solución a una situación y donde el control del proceso de negocio

está caracterizado por una de las partes que intervienen en el proceso. Por otra parte, la coreografía como una secuencia de mensajes que pueden involucrar múltiples participantes y múltiples fuentes, incluyendo clientes, proveedores y socios. La coreografía es la colaboración de los participantes involucrados en el proceso de negocio desde un punto de vista global, un desempeño visible, común y complementario; donde ordenado el mensaje, intercambia un resultado conforme a un objetivo de negocios común.

## **Orquestación como solución a implementar**

La noción de una aplicación o servicio compuesto se basa en la idea de la construcción de nuevas aplicaciones o servicios, interconectando las partes existentes. La orquestación juega un papel importante en esto, ya que es la responsable de aglutinar esas partes al coordinar la ejecución, de cada servicio discreto. Estas actividades son en realidad interacciones entre servicios, es decir, servicios que se invocan unos a otros, pero no de forma desordenada, sino de manera controlada por el orquestador que es quien conoce en detalle de todas las tareas que se deben llevar a cabo para completar el proceso.

Es por estas razones que la solución al planteamiento formulado es este trabajo de grado implementa como composición de servicios la orquestación. A través del inventario de servicios se organizan en capas de servicios, en donde se componen los servicios para el proceso de negocio.

## **Metodología**

Para lograr el desarrollo del presente trabajo de investigación se propone aplicar el método fundamentado en los principios que establece el SOMA (Método de Modelado y Arquitectura Orientados a Servicios, por sus siglas en inglés), propuesto por IBM, que propone como principio fundamental la identificación de servicios conectados con requisitos de negocios.

SOMA consiste en un ciclo iterativo de refinamiento dividido en varias fases, a saber: Modelado del Negocio y Transformación, Gestión de la Solución, Identificación, Especificación, Realización, Implementación, Despliegue y Monitoreo, esto se muestra en la Figura 4. A su vez propone tres pasos para la implementación de los servicios: identificación, especificación y realización de servicios, componentes y flujos. Esto se observa en la Figura 5.



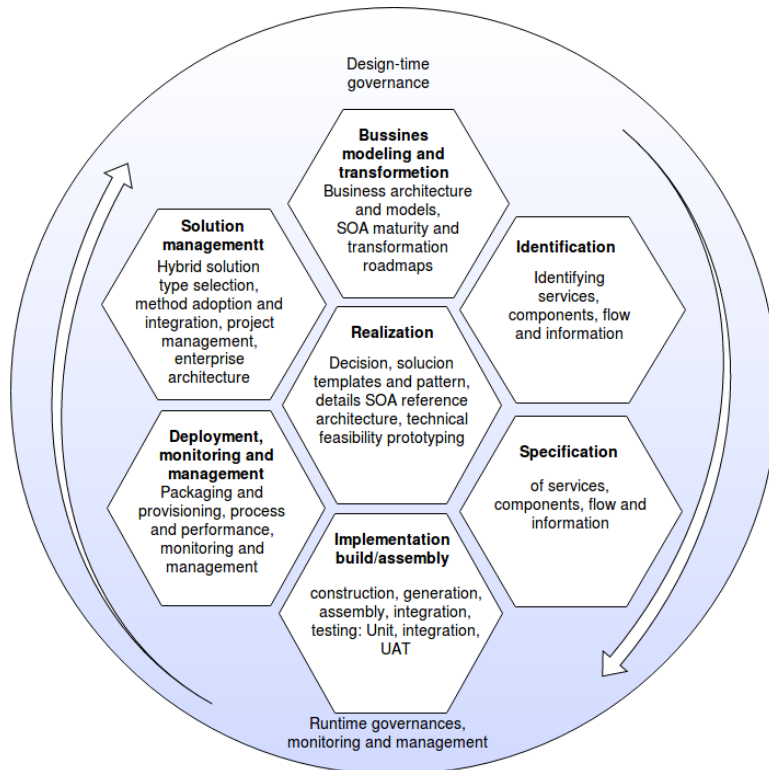


Ilustración 4. Ciclo de vida de SOMA (Arsanjani, 2004).

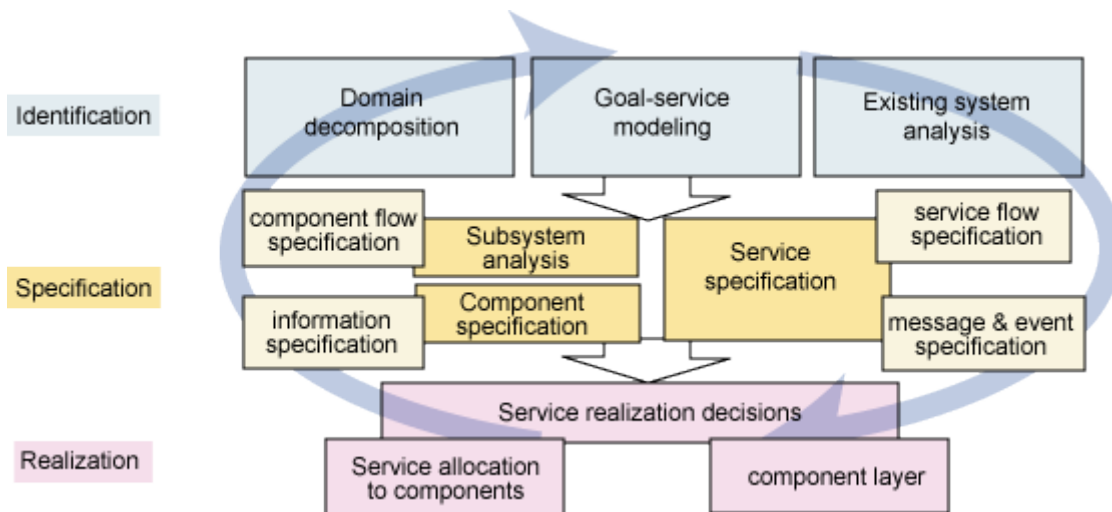


Ilustración 5. Pasos generales de SOMA (Arsanjani, 2004).

Desde el punto de vista conceptual, el método SOMA se basa en un estilo arquitectónico que define un modelo de interacción entre tres partes principales:

1. El proveedor de servicios, que publica una descripción del servicio y proporciona la implementación del servicio.

2. Un consumidor de servicios, que puede utilizar el identificador uniforme de recursos (URI) para vincular e invocar el servicio.
3. Un identificador de servicios, que permite tener una identificación clara y concisa de la descripción del servicio que se proporciona a través de un registro accesible.

En los proyectos basados en el paradigma software como servicios, éstos son identificados a través de enfoques (Sivakumar, G. 2010):

Enfoque **top down**: Tanto la interfaz como la implementación del servicio Web deben ser diseñadas, codificadas y programadas desde cero

Enfoque **bottom up**: En este caso, pueden haber dos posibilidades:

- Existe una aplicación a la cual se le desea crear un servicio Web
  - Implica diseñar la interfaz del servicio Web y crear su descripción WSDL/WADL
- Existe una descripción de la interfaz del servicio web, pero no existe su implementación.
  - Implica diseñar la implementación del servicio web de acuerdo a la descripción de su interfaz

Enfoque **Goal-Service Modeling (GSM)**: Utiliza un enfoque basado en los objetivos del negocio y metas para la identificación de servicios. Combina los enfoques *top down* y *bottom up*.

Para la identificación de los posibles servicios para efectos de este trabajo la técnica a emplear será el enfoque *Goal Service Modeling*. Esto motivado a que da una visión intermedia entre los enfoques *top down* y *bottom up*, consistiendo en el modelado de servicios vinculando los servicios con los objetivos del negocio.

Para aplicar el método SOMA en la solución del problema planteado, primero se identificará el problema, seguidamente se describirá el objeto de estudio a ser abordado, para posteriormente iniciar el desarrollo de la herramienta. Durante este proceso, y para cumplir las etapas de SOMA, se identificarán los posibles servicios, realizando una categorización o jerarquía de los servicios candidatos, que de este modo, permita determinar la composición y construcción de servicios interdependientes.

Posteriormente el método propone definir el proceso de negocio a través de la notación de modelado de procesos de negocio (BPMN, por sus siglas en inglés) para la sistematización del negocio y lograr la gestión del flujo de trabajo y tareas asociadas, esta fase ya fue realizada y se muestra en la Figura 1.

Seguidamente, se definirán modelos de diseño, utilizando para ello el Lenguaje de Modelado Unificado (UML), donde cada uno permita presentar una visión o perspectiva de dicho sistema, que brinde la especificación de los componentes. Igualmente, se implementará la especificación de requisitos según el estándar de

IEEE 830 que ayude a organizar de manera estructurada los requisitos del sistema.

El siguiente paso es la implementación de los servicios a través del lenguaje de programación java, ya identificados y diseñados con las respectivas técnicas, se realizó la generación de la interfaz del servicio web y codificación de la implementación del servicio web.

Lo siguiente es la elaboración del plan de pruebas. Diseñar, preparar y ejecutar las pruebas funcionales y no funcionales para la validación y verificación del servicio web. De encontrarse errores, corregir los errores encontrados durante las pruebas.

Por último, una vez desarrollado el sistema de conciliación de la tesorería internacional del BCV, se someterá a operación y ajustes respectivos, si este fuera el caso. Desplegando el paquete de servicios en ambiente de producción se realizará una pequeña prueba de la aplicación para validar su correcto funcionamiento. Se establece un lapso de monitorización de los servicios para chequear el rendimiento del sistema y el flujo del proceso.

# CAPITULO III

## Marco Aplicativo

### Fase 1. Modelado del negocio y transformación

Durante esta fase se hizo el estudio y análisis del Departamento de Inversiones y Tesorería Internacional y su relación con los Departamentos del instituto del BCV. Adicionalmente se hizo una investigación de las mejores prácticas para la implementación de una arquitectura orientada a servicios y principios de diseño, de las herramientas y tecnologías que se utilizaron en el proyecto.

Se hizo el análisis de requerimiento y la especificación de requisitos según el estándar de IEEE 830 para el proceso de negocio de la conciliación de la tesorería internacional y el modelado del negocio en BPMN.

El modelo de proceso de negocio va a ser utilizado para describir, en primer término, y entender el proceso de negocio, así como el entorno en el que se va a integrar el sistema de información que se tiene como objetivo del proceso completo de desarrollo.

Con el fin de modelar este proceso se ha adoptado la notación para el modelado de procesos de negocio (Business Process Modeling Notation, BPMN). La principal razón para tomar esta decisión ha sido que BPMN se ha convertido en el estándar en la industria para la definición de procesos de negocio, y su utilización está muy extendida entre los profesionales implicados en este ámbito, con lo que será posible aprovechar la mayoría del conocimiento sobre definición de procesos de negocio que ya tienen las organizaciones, adicionalmente, como se hizo mención en el capítulo II el método SOMA se complementa fundamentalmente de BPMN, con lo que para la identificación, especificación y realización de los servicios es base para su desarrollo.

En este primer paso del método para la especificación de requisitos se pretende construir modelos de un alto nivel de abstracción, con los cuales permita analizar y desarrollar una arquitectura de servicios donde se pueda identificar los servicios que deben ser ofrecidos al usuario final del sistema de información. Éstos pueden identificar servicios a partir de los diferentes subprocesos que se definan dentro del propio proceso de negocio.

Durante esta fase se realizaron reuniones con los usuarios funcionales donde se estableció los siguientes criterios:

- **Alcance del proceso de negocio a implementar:** El ámbito del proceso abarca la recepción y procesamiento de los mensajes SWIFT, su respectiva conciliación con las solicitudes generadas por los departamentos y generación de los reportes. Todo en conjunto con el manejo de los flujos de fondos.
- **Roles:** Los roles involucrados en el proceso de negocio son el Analista Financiero quien tiene la potestad de consultar toda la información referente al proceso y el Coordinador de Operaciones Internacionales quien aparte de tener los privilegios del Analista Financiero tiene la responsabilidad de administrar el sistema (reportes, configuraciones del sistema, etc...).
- **Especificación de los requerimientos:** Durante las reuniones de trabajo con los usuarios funcionales se establecieron requerimientos específicos. A continuación se mencionan algunos.
  - Recepción y procesamiento de los mensajes SWIFT
  - Recepción de las solicitudes de coberturas de los departamentos involucrados
  - Cotejamiento de los balances recibidos de los corresponsales con las coberturas solicitadas
  - Generación de reportes del proceso de conciliación
  - Notificación por correo electrónico de las operaciones no identificadas y no ejecutadas
  - Creación de un catálogo de instrumentos para el manejo de los flujos de fondo

En la tabla 1 se describe el proceso de conciliación de la tesorería internacional (ver figura 1 para el modelado de negocio en BPMN)

*Tabla 1 Descripción del proceso de conciliación de la Tesorería Internacional*

<b>Nombre proceso</b>	Conciliación
<b>Descripción</b>	El proceso de conciliación es el cotejamiento de los balances de los corresponsales con lo que cuenta el BCV contra las solicitudes de coberturas (crédito/debito) de divisas que realizan los departamentos quienes interactúan con el DITI
<b>Inicio</b>	Se recibe los balances SWIFT MT949/MT950 emitidos por los corresponsales en el extranjero. Se revisa cada operación hecha en la cuenta corriente junto con las solicitudes de coberturas hechas por los departamentos
<b>Finaliza</b>	Con la comparación de las operaciones se obtiene la conciliación y se procede a la afectación de los fondos obteniendo finalmente la posición de los fondos de cada corresponsal con sus respectivos instrumentos o tramos. Se identifica las operaciones que no fueron ejecutadas y se precisa las operaciones que no fueron identificadas remitiendo las mismas a los departamentos

	para que indiquen la procedencia de esas operaciones.
<b>Entradas</b>	Solicitudes de coberturas y Balances MT950/MT940
<b>Salidas</b>	Posición de los fondos, conciliación, notificación de los montos no identificados y no ejecutados.
<b>Consumidor del proceso</b>	Sistema de régimen de administración de divisas SRAD, Liquidador Genérico
<b>Expectativa del cliente del proceso</b>	Las expectativas del usuario final están centradas en los siguientes puntos: <ul style="list-style-type: none"> <li>• Obtener y manejar el flujo de fondos</li> <li>• Administrar el catálogo de instrumentos</li> <li>• Captar las solicitudes de coberturas</li> <li>• Recibir y procesar los balances de los corresponsales</li> <li>• Generar los reportes de las operaciones no identificadas y no ejecutadas</li> </ul>

Las actividades identificadas como tareas humanas constituyen la interfaz gráfica de usuario (GUI, por sus siglas en inglés) o *front end* que estarán hecho en una aplicación web. Para las tareas que no son humanas y que son de negocio repetitivas serán ejecutadas por el sistema de conciliación y constituyen invocaciones a servicios.

La identificación de estas funcionalidades de negocio repetitivas y comunes permitió definir los candidatos a servicios de la arquitectura. Junto con el modelado del proceso de negocio, se recogieron los requerimientos funcionales que fueron obtenidos con base a la especificación de requisitos según el estándar de IEEE 830.

## **Especificación de Requisitos**

### **Proceso de Conciliación de la Tesorería Internacional del BCV**

#### **Introducción**

El Departamento de Inversiones de Tesorería Internacional (DITI) de la Gerencia de Administración de Reservas Internacionales (GARI) del Banco Central de Venezuela (BCV) requiere de una solución tecnológica que permita automatizar, controlar y gestionar todo el **proceso de conciliación de saldos de los corresponsales** con los que trabaja; Permitiéndole, con ello, agilizar y optimizar la gestión de los recursos líquidos (cash) y generar la posición de saldos de cada corresponsal, y ayudando, así, dar respuesta de manera oportuna a los demás departamentos del Instituto y adicionalmente dar información rápida y verídica a las autoridades para la toma de decisiones.

## Propósito

A través del presente especificación de requisitos se establecerá los requerimientos del cliente y/o de los usuarios finales y la especificación de requisitos del sistema con la finalidad de delimitar el compromiso que se adquirirá para el desarrollo del sistema. Tales descripciones permitirá indicar lo que debe hacer el sistema: Servicios que ofrecerá, las restricciones en su operación y las metas propuestas del software. Los requerimientos reflejarán las necesidades del cliente para la automatización del **proceso de conciliación de operaciones de la tesorería internacional**.

## Ámbito del Sistema

El sistema recibirá y procesará los balances emitidos por los corresponsales a través de los mensajes SWIFT (The Society for Worldwide Interbank Financial Telecommunications, por sus siglas en inglés) desde donde realizará la comparación de los saldos ejecutados contra los saldos notificados. Generará el reporte con las operaciones con y sin conciliar:

- Las operaciones notificadas y que no fueron ejecutadas serán base para el reporte de **“movimientos no ejecutados (MNE)”** con montos de operaciones no ejecutadas pero que fueron estimadas.
- Las operaciones ejecutadas y no notificadas serán base para el reporte de los saldos de **“movimientos no identificados (MNI)”** con los montos de operaciones ejecutadas que no estaban previstas.

Según sea requerido, serán remitidos al Departamento de Corresponsalía o a los distintos Departamentos del Instituto.

Luego de ello, se recibirá, revisará y realizará los ajustes correspondientes generando nuevamente los reportes de saldos de los balances: montos reales y estimados. De continuar reparos; se ejecutará nuevamente el proceso de conciliación.

El DITI contará con el sistema propuesto que automatizará los procedimientos necesarios para la generación de las conciliaciones, permitiendo elaborar y remitir de forma ágil los reportes e informes de los saldos de los balances, monto reales y estimados con el propósito de tener los datos esenciales para la toma de decisiones y medición de resultados de la gestión de los recursos líquidos del instituto.

Es decir, el **objetivo general es desarrollar y proveer un sistema de conciliación que genere una herramienta de gestión efectiva y eficiente de los flujos de ingreso y egresos de divisas y de la previsión y control de los movimientos de saldos**, con el objetivo de garantizar de manera oportuna la

colocación de excedentes y la facilitación de recursos a otras unidades del instituto para el cumplimiento de obligaciones, a los fines de convertir la tesorería internacional en un departamento mucho más eficiente y productivo, mediante la reducción de costos financieros y la provisión oportuna y veraz de información a las autoridades para la toma de decisiones.

## **Definiciones, Acrónimos y Abreviaturas**

**Tramo:** Subdivisión dentro de un conjunto total de activos que conforma un portafolio, y que responde a criterios de inversión tales como liquidez, riesgo y/o rentabilidad.

**MT940, MT950:** Son tipos de mensajes SWIFT en los que se remite los estados de las cuentas de los corresponsales.

**Arquitectura de Software:** Muestra la distribución de funciones a través de los módulos del sistema. Entonces, El diseño arquitectónico se interesa por entender cómo debe organizarse un sistema y cómo tiene que diseñarse la estructura global de ese sistema.

**Interfaz de comunicación:** Representa la capacidad de comunicación entre componentes de software internos o externos a través de estándares de comunicación. Ésta interfaz define los datos disponibles y cómo se puede acceder a ellos.

## **Descripción general**

El desarrollo del producto de software será implementado con tecnologías abiertas, multiplataforma e interoperable para así cumplir con la legislación nacional y las políticas emanadas por la Gerencia de Sistema e Informática del BCV.

## **Perspectiva del producto**

El producto de software será base de un sistema mayor que cubra otros procesos de negocio de la DITI carentes de un sistema automatizado que le permita agilizar el trabajo; y será interoperable con otros sistemas que existen en el BCV. Con relación a esto último, coexistirá y se relacionará con el Sistema Automatizado de Tesorería Internacional (SATI) a quien le proveerá los datos ya conciliados de las posiciones del día de los corresponsales.

## **Funciones del producto**

### **1. El usuario podrá cargar en el sistema parámetros de configuración:**

Se permitirá al usuario con el rol de Coordinador de Operaciones Internacionales la configuración del sistema en referencia a:



- Indicar los correos que serán utilizados para la notificación de las operaciones no identificadas o no ejecutadas.
- Colocar el mensaje y asunto que llevará el correo
- Establecer la hora en que se notificaran las operaciones
- Establecer la hora de conciliación
- Configurar los parámetros de sistema

## 2. Recibir mediante el estándar de comunicación SWIFT los mensajes de los balances de los corresponsales:

El sistema deberá soportar los estándares de comunicación SWIFT permitiendo recibir, indistintamente la cantidad, los mensajes de estados de cuenta que son emitidos por los distintos corresponsales.

## 3. Almacenar los balances de los saldos recibidos de los corresponsales:

A través de un manejador de base de datos guardar los balances recibidos de los corresponsales.

Para cumplir con el requerimiento se propone implementar una estructura con base al análisis hecho de la siguiente manera:

- **RIO (Reservas Internacionales Operativas) / OAME (Otros Activos en Moneda Extranjera)**
  - ✓ **PROPIETARIO - ORIGEN-FUENTE (Portafolio – Cartera)**
    - **TRAMO**
      - ❖ Tipo instrumento
      - ❖ Banco
      - ❖ cuenta corriente
      - ❖ Plaza
      - ❖ Moneda
      - ❖ Mercado – Tipo de Cambio
      - ❖ Cuenta contable
      - ❖ Operación de negocio
      - ❖ Monto
      - ❖ Fecha

### *Ilustración 6 Propuesta de estructuración de la cartera*

**RIO:** Son los activos líquidos con los que cuenta el banco para la inversión y cumplimiento a los compromisos contraídos.

**OAME:** Otros activos líquidos expresados en otras monedas extranjeras distintas al dólar estadounidense y fondos de otros entes públicos que son administrados por el BCV.

**Propietario:** Son los orígenes de la fuente del dinero.

**Tramo:** Subdivisión dentro de un conjunto total de activos que conforma un portafolio.

**Banco:** Corresponsal dedicada a realizar operaciones financieras en representación del Banco Central.

**Moneda:** Instrumento representativo del dinero circulante en cada país.

**Mercado/Tipo de cambio:** El tipo de cambio representa el precio de una moneda en términos de otra moneda. En el caso venezolano, se refiere a la cantidad de bolívares por un dólar estadounidense.

**Operación:** Es aquella transacción de divisas de ingreso o egreso cuyo acuerdo de negociación se realiza para que se liquide en un plazo determinado.

**Monto:** Es la cantidad de moneda que se tiene en la operación.

**Fecha:** Es en relación a la fecha de operación o de valor donde según sea el caso se pactan o se hace efectiva las operaciones en el mercado de divisas.

**4. Generar reporte con los saldos ejecutados contra los saldos notificados:**

A través de los archivos SWIFT hacer la comparación de los saldos ejecutados y los saldos notificados a través de las solicitudes de coberturas. A partir de dicha comparación generar un reporte que refleje los saldos que concuerdan y los saldos que no son identificados.

**5. Permitir consultar a través de un dato que sea inequívoco los reportes de conciliación que son generados por el sistema:**

Mediante un dato inequívoco el sistema le devolverá el reporte de conciliación realizado por el sistema en un determinado momento al usuario.

**6. El usuario tendrá la posibilidad de generar reportes según lo establecido en la configuración hecha por el rol de Coordinador de Operaciones Internacionales del sistema:**

Entre los reportes que se puedan generar están por los ingresos o egresos de los instrumentos (BCV/FEM/OAME), saldo por países, saldos por corresponsales, etc...

**7. Remitirá un correo electrónico a los distintos departamentos del instituto con los montos no identificados:**

El sistema deberá remitir un correo a los departamentos con los saldos no identificados en la conciliación para que den respuesta a través de un servicio si corresponden o no esos fondos a una ejecución en su departamento.

## **8. Administración de los corresponsales e instrumentos**

Permitirá crear corresponsales con sus datos asociados y hacer el manejo lógico de esos fondos a través de un catálogo de instrumentos

## **9. Permitir consultar los movimientos generados por la afectación de los saldos**

A través de un dato inequívoco realizar consultas de los movimientos de la afectación de los saldos de los corresponsales y de los instrumentos

### **Características de los usuarios**

Los usuarios del sistema incluyen a Coordinador de Operaciones Internacionales II, Especialistas Financieros I y II, Técnicos; Analistas Financieros I II y III; todos ellos tienen al menos un grado académico de licenciado o equivalente. La mayoría tiene experticia en el proceso de conciliación bancaria.

Todos están adscritos al Departamento de Tesorería Internacional y el sistema va enfocado para dichos usuarios.

### **Restricciones**

La Gerencia de Sistemas e Informática del BCV tiene lineamientos, criterios, normas y métodos de trabajo para la estandarización de desarrollo de un software con el fin de establecer la calidad de los sistemas hechos por las unidades de dicha gerencia. Adicionalmente, en cuanto a las consideraciones de seguridad del software deberán ser mediante un método de autenticación y autorización mediante un sistema de verificación de credenciales a través de LDAP. La responsabilidad de la seguridad recaerá a nivel de red y/o servidores, por lo que el software propuesto no implementará mayor nivel de seguridad que el antes mencionado.

El lenguaje de programación a ser usado para el desarrollo del software será Java EE (JAVA Enterprise Edition, por sus siglas en inglés). Permitirá desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Permitiendo utilizar arquitecturas de N capas distribuidas, y Arquitecturas Orientadas a servicios (SOA) y apoya ampliamente el uso de componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

### **Suposiciones y dependencias**

Entre los factores dependientes, está el uso de SWIFT, en caso de ser cambiado en un futuro, por otro estándar, esto afectará los requisitos por lo que deberá ser revisado y/o cambiado los requisitos aquí plasmados.

## Requisitos específicos

*Tabla 2 Requerimientos del sistema*

Definición del requerimiento del usuario	
I	El sistema captará diariamente los balances de saldos emitidos electrónicamente por los corresponsales indistintamente de la cantidad emitida.
Especificación de los requerimientos del sistema	
I.I	El software recibirá a través del protocolo de comunicación SWIFT cada mensaje de tipo MT940 y MT950 los estados de los balances de las cuentas de los corresponsales
I.II	Interpretará e identificará cada mensaje SWIFT (banco emisor, tipo de mensaje, prioridad, hora de envío y hora de recepción. (ver tabla 1)
I.III	Extraerá del mensaje SWIFT las operaciones débito y crédito por fecha valor

*Tabla 3Requerimientos del sistema*

Definición del requerimiento del usuario	
I	Guardará los datos de los mensajes SWIFT para tener un histórico
Especificación de los requerimientos del sistema	
II.I	Los datos SWIFT procesados serán almacenados en un manejador de base de datos
II.II	Estructurará y clasificará los datos por corresponsal en la base de datos en función de que pueda ser recuperada de manera rápida y organizada
II.III	El analista podrá consultar los balances recibidos por fecha valor

*Tabla 4Requerimientos del sistema*

Definición del requerimiento del usuario	
I	El sistema elaborará informes diarios que revelen la discrepancia o concordancia(proceso de conciliar) los saldos ejecutados contra los saldos notificados
Especificación de los requerimientos del sistema	
III.I	Diariamente el sistema automáticamente generará un reporte de las operaciones realizadas. (Ver punto III.III) saldos de los balances, montos

	reales y estimados por fecha valor
III.II	El sistema generará el reporte en formato Excel.
III.III	Se realizará un reporte por cada corresponsal donde estén los atributos a ser: <ul style="list-style-type: none"> <li>• Nombre del corresponsal</li> <li>• Identificación del instrumento</li> <li>• Fecha Valor</li> <li>• Moneda</li> <li>• Monto</li> <li>• Débito/crédito</li> <li>• Observaciones</li> </ul>
II.IV	El acceso a los informes estarán vinculados exclusivamente a usuarios autorizados en la lista de control de acceso

*Tabla 5Requerimientos del sistema*

Definición del requerimiento del usuario	
IV	Las partidas no identificadas deberán ser enviado a los departamentos para su reconocimiento
Especificación de los requerimientos del sistema	
IV.I	Se remitirá vía correo electrónico a todos los departamentos las operaciones no identificadas

*Tabla 6Requerimientos del sistema*

Definición del requerimiento del usuario	
V	Las partidas que no fueron ejecutadas deberán enviarse a corresponsalía
Especificación de los requerimientos del sistema	
V.I	Se remitirá vía correo electrónico a corresponsalía las operaciones que no fueron ejecutadas

*Tabla 7Requerimientos del sistema*

Definición del requerimiento del usuario	
VI	Recibir por parte de los distintos departamentos respuesta de las operaciones no identificadas; de no recibir a tiempo dicha respuesta se remitirá a corresponsalía

Especificación de los requerimientos del sistema	
VI.I	A través de un servicio se captará la adjudicación de la operación no identificada según el departamento
VI.II	Transcurrido una hora determinada por la configuración del usuario con privilegios de administrador; las operaciones no identificadas se remitirán a corresponsalía

*Tabla 8 Requerimientos del sistema*

Definición del requerimiento del usuario	
VII	El sistema realizará ajustes por cada partida ya identificada según el departamento y lo relacionará al corresponsal y/o instrumento correspondiente
Especificación de los requerimientos del sistema	
VII.I	Añadirá la operación ya identificada al instrumento que le corresponda
VII.II	Ejecutará la conciliación de las operaciones

*Tabla 9 Requerimientos del sistema*

Definición del requerimiento del usuario	
VIII	El sistema deberá permitir generar el reporte de los balances, montos reales y estimados en cualquier momento que el analista lo requiera
Especificación de los requerimientos del sistema	
VIII.I	A través de una interfaz gráfica de usuario (GUI) el usuario autenticado y con privilegio podrá generar el reporte de los balances, montos reales y estimados por fecha valor

*Tabla 10 Requerimientos del sistema*

Definición del requerimiento del usuario	
IX	Los saldos conciliados deben ser almacenados en base de datos y afectar el flujo de fondos
Especificación de los requerimientos del sistema	
IX.I	Las operaciones conciliadas serán registradas en la base de datos generando una afectación en los saldos

*Tabla 11 Requerimientos del sistema*

Definición del requerimiento del usuario	
X	El sistema debe permitir establecer un mensaje y asunto configurable para el email
Especificación de los requerimientos del sistema	
X.I	Los usuarios autenticados y con privilegios de administrador podrán cambiar los atributos por defecto del mensaje y asunto del correo

*Tabla 12 Requerimientos del sistema*

Definición del requerimiento del usuario	
XI.	Deberá permitirse la administración de los corresponsales y del catálogo de los instrumentos
Especificación de los requerimientos del sistema	
XI.I	Los usuarios autenticados podrán crear y modificar los corresponsales con sus respectivos datos
XI.II	Permitirá crear y modificar instrumentos del catálogo de instrumentos

*Tabla 13 Requerimientos del sistema*

Definición del requerimiento del usuario	
XII	Consultar los movimientos generados de las operaciones a través de un dato inequívoco
Especificación de los requerimientos del sistema	
XII.I	Usando la fecha valor como rango de búsqueda extraer los movimientos generados en esa fecha

### **Atributos del sistema**

- **Adecuación Funcional:** El producto de software debe proporcionar las funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas en el diseño.
- **Compatibilidad:** El sistema deberá poder intercambiar información y/o llevar a cabo sus funciones requeridas cuando deba comunicarse con otro sistema externo.

- **Usabilidad:** El producto de software debe ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se use bajo determinadas condiciones.
- **Fiabilidad:**
  - **Disponibilidad:** El sistema debe ser capaz de estar operativo y accesible para su uso cuando se requiera

## Fase 2. Identificación

Para esta fase se identificaron algunos servicios candidatos a través de las actividades del proceso de negocio, junto a componentes y flujos del proceso. Para cada servicio identificado se relacionó que requerimientos deberían satisfacer, esto con la finalidad de obtener nuevas operaciones que los servicios deberían satisfacer, así como requerimientos no funcionales.

### Casos de uso

Se usa la modelación de casos de uso en proyectos de tecnologías de la información para capturar los requisitos funcionales de un sistema. Los modelos de caso de uso se basan en conceptos que soportan la reutilización, los cuales deben ajustarse bien a una arquitectura como SOA, que promueve la reutilización. La mayoría de las técnicas de modelación de caso de uso utilizadas se basan en conceptos que son anteriores a SOA y a la gestión de procesos empresariales (BPM). (Evans, M. y Schoonderwoerd, R., 2011)

Los modelos de caso de uso que se desarrolla con la técnica provista por el método SOMA pueden permanecer válidos y útiles más allá de este proyecto. Los modelos de caso de uso se pueden usar como referencia para nuevos proyectos que buscan la reutilización más allá de la duración de este proyecto de entrega. Frecuentemente los casos de uso resultantes son análogos a servicios; por tanto, la especificación de los casos de uso es parte importante para la especificación de servicios.

Los casos de uso para el proceso de conciliación describen la secuencia normal del proceso de negocio, secuencias alternativas, reglas de negocio, precondiciones y postcondiciones.

### Modelación de casos de uso alineados a SOA

La siguiente figura muestra la estructura correspondiente al caso de uso. Cada caso de uso define un proceso correspondiente con más detalles. También define las características del servicio que se usará para invocar ese proceso. Por ejemplo: las precondiciones y pos condiciones, entradas y salidas del caso de uso definen las dependencias del servicio, los límites y los requisitos de datos.



Es importante señalar que los casos de usos a continuación presentados representan las principales funcionalidades que va a proveer los servicios identificados, esto para reducir la cantidad de páginas en el documento y la misma no sea excesivamente extenuante para el lector.

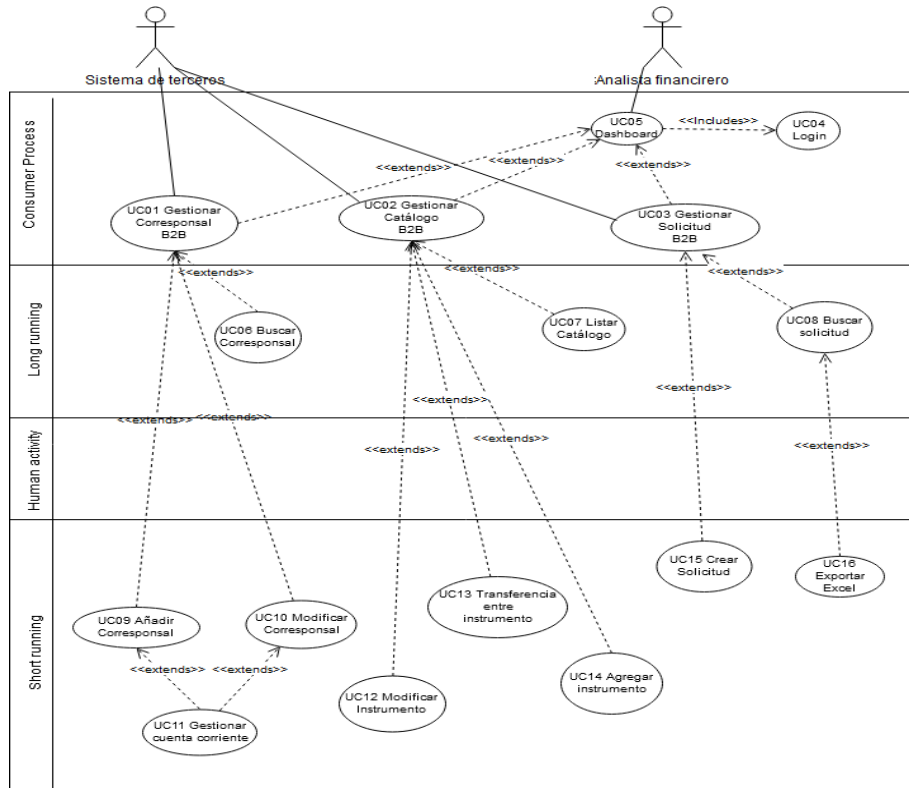


Ilustración 7 Casos de Uso. Proceso de conciliar operaciones

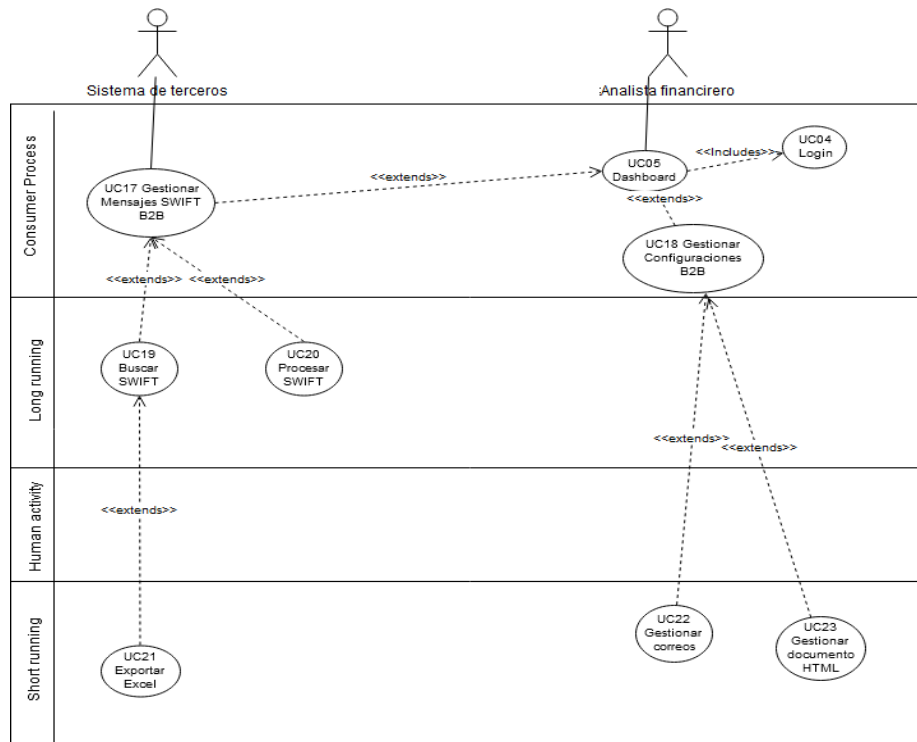


Ilustración 8 Casos de Uso. Proceso de conciliar operaciones

### Capa de proceso del consumidor

Esta sección repasa y describe los casos de uso numerados en el proceso del consumidor de la figura 7 y 8.

### Caso de uso UC05: Dashboard

El analista financiero al ingresar al panel de control podrá examinar las distintas opciones disponibles a gestionar.

#### Meta

Gestionar alguno de los ítems disponibles entre las opciones.

#### Desencadenante

El ingreso satisfactorio al panel de control.

#### Actor principal

Analista financiero.

#### Precondiciones

El inicio de sesión fue correcto.

#### Post-condiciones

Aparece un menú de navegación donde lista los ítems disponibles.

### **Descripción del caso de uso**

La aplicación web muestra al analista las opciones de navegar entre los ítems del menú en el panel de control. El caso de uso finaliza cuando el analista selecciona algún ítem del menú.

### **Incluye**

UC04 Login

### **Caso de uso UC01: Gestionar corresponsal B2B**

Soporta un canal de gestión de corresponsales para procesos de negocios a través de una interfaz de servicio business to business (B2B)

### **Meta**

Gestionar los corresponsales a través de una interfaz de servicio.

### **Desencadenante**

Un proceso de negocio hace la petición de una operación del servicio del corresponsal.

### **Actor principal**

Sistema de terceros, sistema de conciliación.

### **Precondiciones**

Ninguno

### **Post-condiciones**

Según la operación que se consuma, añadirá, listará o modificará un corresponsal.

### **Descripción de caso de uso**

El canal de servicio provee las operaciones para gestionar los corresponsales.

### **Extiende**

UC06 añadir corresponsal, UC09 buscar corresponsal y UC10 modificar corresponsal

### **Caso de uso UC02: Gestionar catálogo B2B**

Soporta un canal de gestión de la cartera de instrumentos para procesos de negocios a través de una interfaz de servicio business to business (B2B).

**Meta**

Gestionar el catálogo de instrumentos a través de una interfaz de servicio.

**Desencadenante**

Un proceso de negocio hace la petición de una operación del servicio de catálogo.

**Actor principal**

Sistema de terceros, sistema de conciliación.

**Precondiciones**

Ninguno.

**Post-condiciones**

Listará todo el catálogo de instrumentos.

**Descripción de caso de uso**

Permite a través del canal de servicio gestionar el catálogo de instrumentos.

**Incluye**

UC07 listar catálogo, UC12 modificar instrumento, UC13 transferencia entre instrumentos, UC14 agregar instrumento

**Caso de Uso UC03: Gestionar solicitud B2B**

Soporta un canal de gestión de las solicitudes de coberturas para procesos de negocios a través de una interfaz de servicio business to business (B2B).

**Meta**

Gestionar las solicitudes de coberturas a través de una interfaz de servicio.

**Desencadenante**

Un proceso de negocio hace la petición de una operación del servicio de solicitudes.

**Actor principal**

Sistema de terceros, sistema de conciliación.

**Precondiciones**

Ninguno

**Post-condiciones**

Permite crear una solicitud de cobertura o hacer la búsqueda de las solicitudes.

### **Descripción de caso de uso**

Permite a través del canal de servicio gestionar las solicitudes de coberturas de las áreas de negocio.

### **Extiende**

UC08 buscar solicitud, UC15 crear solicitud

## **Caso de Uso UC17: Gestionar mensajes SWIFT B2B**

Soporta un canal de gestión de los balances SWIFT para procesos de negocios a través de una interfaz de servicio business to business (B2B)

### **Meta**

Gestionar los balances SWIFT a través de una interfaz de servicio.

### **Desencadenante**

Un proceso de negocio hace la petición de una operación del servicio de mensajes SWIFT.

### **Actor principal**

Sistema de terceros, sistema de conciliación.

### **Precondiciones**

Ninguno

### **Post-condiciones**

Permite disparar el proceso de conciliación de operaciones o hacer la búsqueda de los balances SWIFT y exportarlos a Excel.

### **Descripción de caso de uso**

Permite a través del canal de servicio gestionar los balances SWIFT de los corresponsales.

### **Extiende**

UC19 buscar SWIFT, UC20 procesar SWIFT

## **Caso de Uso UC18: Gestionar configuraciones B2B**

Soporta un canal de gestión de la configuración de la aplicación web para el proceso de conciliación a través de una interfaz de servicio business to business (B2B).

### **Meta**

Gestionar las configuraciones a través de una interfaz de servicio.

### **Desencadenante**

La aplicación web hace la petición de una operación del servicio de configuraciones.

### **Actor principal**

Analista financiero.

### **Precondiciones**

Ninguno

### **Post-condiciones**

Permite gestionar los correos que se usaran para las notificaciones de operaciones no identificadas ni ejecutadas, adicionalmente, permite establecer un asunto y contenido al emitir el correo por el proceso de conciliación.

### **Descripción de caso de uso**

Permite a través del canal de servicio gestionar los correos electrónicos y el asunto y contenido del correo. El correo debe contener un formato válido, de no ser así, se emite una respuesta de error a quien hizo el llamado de la operación.

### **Extiende**

UC22 Gestionar Correo, UC23 Gestionar Documento HTML.

## **Capa de proceso de larga duración**

Esta sección repasa y describe los casos de uso numerados en el proceso de larga duración de la figura 8 y 9.

## **Caso de Uso UC06: Buscar corresponsal**

Cuando una operación del servicio es requerido (por el caso de uso Gestionar Corresponsal), se inicia el proceso de búsqueda de corresponsal. Se trata de un proceso de larga duración.

### **Meta**

Enviar el listado de corresponsales en respuesta a la invocación del servicio que se hizo.

### **Desencadenante**

Se invoca una operación del servicio.

### **Actor principal**

N/D

### **Precondiciones**

Es necesario que se haya hecho una solicitud de invocación de la operación del servicio.

### **Post-condiciones**

El criterio de búsqueda de los corresponsales debe haber sido suministrado. Se envió como respuesta los elementos contentivos de los corresponsales.

O bien el criterio de búsqueda no fue suministrado o los datos suministrados son inexistentes. Si cualquiera de ellos permanece sin resolución, el actor recibe una respuesta acorde al caso hasta que cumpla las condiciones del caso de uso.

### **Descripción de caso de uso**

Ese caso de uso representa un proceso de larga duración que orquesta los servicios que ofrecen la información relacionada al corresponsal. Si con los criterios de búsqueda encuentra corresponsales, se invoca los servicios de que provee la cuenta corriente, la moneda ISO y las plazas relacionados a los corresponsales. Luego se envía el listado de los corresponsales con su información al actor invocante del caso de uso.

### **Extiende**

UC10 Modificar corresponsal.

### **Caso de Uso UC07: Listar catálogo**

Cuando una operación del servicio es requerido (por el caso de uso Gestionar Catálogo), se inicia el proceso de búsqueda del catálogo de instrumentos. Se trata de un proceso de larga duración.

### **Meta**

Enviar el catálogo de instrumentos en respuesta a la invocación del servicio que se hizo.

### **Desencadenante**

Se invoca una operación del servicio.

**Actor principal**

N/D

**Precondiciones**

Es necesario que se haya hecho una solicitud de invocación de la operación del servicio.

**Post-condiciones**

Se envía como respuesta los elementos jerárquicos contentivos de los instrumentos.

**Descripción de caso de uso**

El caso de uso representa un proceso de larga duración que orquesta los servicios que ofrecen la información relacionada al catálogo de instrumentos. Se invoca los servicios que provee los corresponsales, tipos de instrumento y los tipos de mercado relacionados a los instrumentos. Luego se envía los elementos jerárquicos de los instrumentos con su información al actor invocante del caso de uso.

**Extiende**

UC12 Modificar instrumento, UC14 Agregar instrumento

**Caso de Uso UC08: Buscar solicitud**

Cuando una operación del servicio es requerido (por el caso de uso Gestionar Solicitud), se inicia el proceso de búsqueda de las solicitudes de coberturas. Se trata de un proceso de larga duración.

**Meta**

Enviar el listado de solicitudes de cobertura en respuesta a la invocación del servicio que se hizo.

**Desencadenante**

Se invoca una operación del servicio.

**Actor principal**

N/D

**Precondiciones**

Es necesario que se haya hecho una solicitud de invocación de la operación del servicio.



## **Post-condiciones**

El criterio de búsqueda de las solicitudes de cobertura debe haber sido suministrado. Se envió como respuesta los elementos contentivos de las solicitudes de cobertura.

O bien el criterio de búsqueda no fue suministrado o los datos suministrados son inexistentes. Si cualquiera de ellos permanece sin resolución, el actor recibe una respuesta acorde al caso hasta que cumpla las condiciones del caso de uso.

## **Descripción de caso de uso**

El caso de uso representa un proceso de larga duración que orquesta los servicios que ofrecen la información relacionada a las solicitudes de coberturas. Se invoca los servicios que provee las operaciones de negocio y las monedas ISO relacionados a las solicitudes de coberturas. Luego se envía los elementos contentivos con la información de las solicitudes de cobertura al actor invocante del caso de uso.

## **Caso de Uso UC19: Buscar SWIFT**

Cuando una operación del servicio es requerido (por el caso de uso Gestionar SWIFT), se inicia el proceso de búsqueda de los balances SWIFT. Se trata de un proceso de larga duración.

## **Meta**

Enviar el listado de balances SWIFT en respuesta a la invocación del servicio que se hizo.

## **Desencadenante**

Se invoca una operación del servicio.

## **Actor principal**

N/D

## **Precondiciones**

Es necesario que se haya hecho una solicitud de invocación de la operación del servicio.

## **Post-condiciones**

El criterio de búsqueda de los balances SWIFT debe haber sido suministrado. Se envió como respuesta los elementos contentivos de los balances SWIFT.

O bien el criterio de búsqueda no fue suministrado o los datos suministrados son inexistentes. Si cualquiera de ellos permanece sin resolución, el actor recibe una respuesta acorde al caso hasta que cumpla las condiciones del caso de uso.

## **Descripción de caso de uso**

Ese caso de uso representa un proceso de larga duración que ofrece la información relacionada a los balances de los corresponsales. Si con los criterios de búsqueda encuentra los mensajes SWIFT, se envía el listado de los balances con su información al actor invocante del caso de uso.

## **Extiende**

UC21 Exportar Excel

## **Caso de Uso UC20: Procesar SWIFT**

Cuando una operación del servicio es requerido (por el caso de uso Gestionar SWIFT), se inicia el proceso de procesar los balances SWIFT de los corresponsales. Se trata de un proceso de larga duración.

## **Meta**

Descomponer, almacenar y conciliar las operaciones recibida a través los balances SWIFT en respuesta a la invocación del servicio que se hizo.

## **Desencadenante**

Se invoca una operación del servicio

## **Actor principal**

N/D

## **Precondiciones**

Es necesario que se haya hecho una solicitud de invocación de la operación del servicio.

## **Post-condiciones**

El sistema buscará los balances SWIFT provisto a través de un *FileSystem* por el sistema Conector Bilateral, de conseguir archivos SWIFT, se descompone y se almacena sus datos. Seguidamente, se hace la búsqueda de las solicitudes de cobertura y se comparan con las operaciones recibidas a través del balance y se concilian. Se envía como respuesta satisfactoria el correcto proceso de la operación.

O bien no se consigue archivos en el *FileSystem* o quedan operaciones sin conciliar. Si cualquiera de ellos permanece sin resolución, el actor recibe una respuesta acorde al caso.

## **Descripción de caso de uso**

Ese caso de uso representa un proceso de larga duración que ofrece la ejecución del proceso de conciliación. De no existir balances que procesar se le indica al actor que no se han recibidos archivos nuevos. De quedar operaciones sin conciliar se le notifica al actor de la discrepancia de las operaciones.

## **Capa de proceso de corta duración**

Esta sección repasa los casos de uso en el área de corta duración en la parte inferior de la figura 8 y 9.

## **Caso de Uso UC09: Añadir Corresponsal**

Añadir un corresponsal es un caso de uso central en el proceso de conciliación. Este proceso de corta duración se encarga de crear los corresponsales con los que va a estar vinculados al proceso de negocio.

### **Meta**

Validar y almacenar un corresponsal.

### **Desencadenante**

Se invoca la operación del servicio con un corresponsal.

### **Actor Principal**

N/D

### **Precondiciones**

Se debe haber enviado los datos de un corresponsal o más.

### **Post-condiciones**

Se ha almacenado un nuevo corresponsal.

## **Descripción del caso de uso**

Este caso de uso representa un proceso de corta duración que confirma la creación de un nuevo corresponsal. Las validaciones realizadas incluyen los detalles del corresponsal y las cuentas corrientes asociadas en caso de ser suministrado.

Si alguna validación falla, se retorna un mensaje de error al que llama. Si la validación es exitosa, se crea el corresponsal y se retornan sus detalles al que llama. El evento de nuevo corresponsal desencadena el procesamiento subsiguiente de creación de las cuentas corrientes asociadas si los datos fueron suministrados.

## **Extiende**

UC11 Gestionar Cuenta Corriente

### **Caso de Uso UC10: Modificar Corresponsal**

Este proceso de corta duración se encarga de modificar los corresponsales que existen asociados al proceso de negocio.

#### **Meta**

Validar y modificar un corresponsal.

#### **Desencadenante**

Se invoca la operación del servicio con un corresponsal.

#### **Actor Principal**

N/D

#### **Precondiciones**

Se debe haber enviado los datos de un corresponsal o más.

#### **Post-condiciones**

Se ha modificado los datos del corresponsal suministrado.

#### **Descripción del caso de uso**

Este caso de uso representa un proceso de corta duración que confirma la modificación de un corresponsal. Las validaciones realizadas incluyen los detalles del corresponsal y las cuentas corrientes asociadas en caso de ser suministrado.

Si alguna validación falla, se retorna un mensaje de error al que llama. Si la validación es exitosa, se modifica el corresponsal y se retornan sus detalles al que llama. El evento de modificación del corresponsal desencadena el procesamiento subsiguiente de gestión de las cuentas corrientes asociadas si los datos fueron suministrados.

## **Extiende**

UC11 Gestionar Cuenta Corriente

### **Caso de Uso UC14: Agregar Instrumento**

Agregar un instrumento es un caso de uso central en el proceso de conciliación. Este proceso de corta duración se encarga de crear los instrumentos que estarán asociados al proceso de negocio.

#### **Meta**

Validar y crear un instrumento.

**Desencadenante**

Se invoca la operación del servicio con un instrumento.

**Actor Principal**

N/D

**Precondiciones**

Se debe haber enviado los datos de un instrumento o más.

**Post-condiciones**

Se ha creado los datos del instrumento suministrado.

**Descripción del caso de uso**

Este caso de uso representa un proceso de corta duración que confirma la creación de un instrumento y su relación jerárquica del portafolio de instrumentos. Las validaciones realizadas incluyen los detalles del instrumento, el corresponsal y las cuentas corrientes.

Si alguna validación falla, se retorna un mensaje de error al que llama. Si la validación es exitosa, se crea el instrumento y se retornan sus detalles al que llama.

**Caso de Uso UC12: Modificar instrumento**

Este proceso de corta duración se encarga de modificar los instrumentos que existen asociados al proceso de negocio.

**Meta**

Validar y modificar un instrumento.

**Desencadenante**

Se invoca la operación del servicio con un instrumento.

**Actor Principal**

N/D

**Precondiciones**

Se debe haber enviado los datos de un instrumento o más.

**Post-condiciones**

Se ha modificado los datos del instrumento suministrado.

### **Descripción del caso de uso**

Este caso de uso representa un proceso de corta duración que confirma la modificación de un instrumento. Las validaciones realizadas incluyen detalles del instrumento, el corresponsal y las cuentas corrientes asociadas al instrumento.

Si alguna validación falla, se retorna un mensaje de error al que llama. Si la validación es exitosa, se modifica el instrumento y se retornan sus detalles al que llama.

### **Caso de Uso UC13: Transferencia entre instrumento**

Este proceso de corta duración se encarga de traspasar fondos entre instrumentos que existen asociados al proceso de negocio.

#### **Meta**

Validar y traspasar fondos entre instrumentos.

#### **Desencadenante**

Se invoca la operación del servicio con los instrumentos.

#### **Actor Principal**

N/D

#### **Precondiciones**

Se debe haber enviado los datos los instrumentos.

#### **Post-condiciones**

Se ha traspasado los fondos entre los instrumentos.

### **Descripción del caso de uso**

Este caso de uso representa un proceso de corta duración que confirma el traspaso de fondos entre los instrumentos. Las validaciones realizadas incluyen detalles del instrumento, el corresponsal y las cuentas corrientes asociadas al instrumento.

Si alguna validación falla, se retorna un mensaje de error al que llama. Si la validación es exitosa, se traspasa los fondos entre los instrumentos y se retornan sus detalles al que llama.

### **Caso de Uso UC15: Crear solicitud**

Crear una solicitud es un caso de uso central en el proceso de conciliación. Este proceso de corta duración se encarga de crear las solicitudes de coberturas que estarán asociados al proceso de negocio.

#### **Meta**

Validar y crear una solicitud de cobertura.

#### **Desencadenante**

Se invoca la operación del servicio con una solicitud de cobertura.

#### **Actor Principal**

N/D

#### **Precondiciones**

Se debe haber enviado los datos de una solicitud de cobertura o más.

#### **Post-condiciones**

Se ha creado la solicitud de cobertura con los datos suministrados.

#### **Descripción del caso de uso**

Este caso de uso representa un proceso de corta duración que confirma la creación de una solicitud de cobertura. Las validaciones realizadas incluyen los detalles del instrumento, el corresponsal, las cuentas corrientes y la operación del negocio.

Si alguna validación falla, se retorna un mensaje de error al que llama. Si la validación es exitosa, se crea la solicitud de cobertura y se retornan sus detalles al que llama.

### **Caso de Uso UC21: Exportar Excel**

Este proceso de corta duración se encarga de generar el reporte en Excel de los montos no identificados, montos no ejecutados y la posición del flujo de fondos.

#### **Meta**

Generar reporte en Excel.

#### **Desencadenante**

Se invoca la operación del servicio luego de la búsqueda los balances.

#### **Actor Principal**

N/D

**Precondiciones**

Se debe haber enviado los parámetros de búsqueda de los mensajes SWIFT.

**Post-condiciones**

Se genera el reporte en Excel de las operaciones.

**Descripción del caso de uso**

Este caso de uso representa un proceso de corta duración que genera el reporte en Excel de las operaciones no identificadas, no ejecutadas y la posición del flujo de fondos de los corresponsales.

**Caso de Uso UC22: Gestionar Correos**

Este proceso de corta duración se encarga de agregar o eliminar los correos que se usaran para la notificación de los montos no ejecutados y no identificados.

**Meta**

Gestionar correos electrónicos.

**Desencadenante**

Se invoca la operación del servicio gestionar configuraciones.

**Actor Principal**

N/D

**Precondiciones**

Se debe haber invocado la operación de servicio Gestionar Configuraciones.

**Post-condiciones**

Se agrega o elimina un correo electrónico.

**Descripción del caso de uso**

Este caso de uso representa un proceso de corta duración que permite agregar o eliminar los correos que se usaran para notificar las operaciones no ejecutadas y no identificadas.



## Modelo de servicio derivado del Caso de Uso

Todos los procesos debajo de la capa de proceso de consumidor se exponen a través de servicios. Los procesos de consumidor Gestionar Corresponsal B2B, Gestionar Catálogo B2B y Gestionar Solicitud B2B se expone como servicio a sistemas de terceros y a la aplicación web. Se muestra el modelo de servicio resultante en la Figura 9.

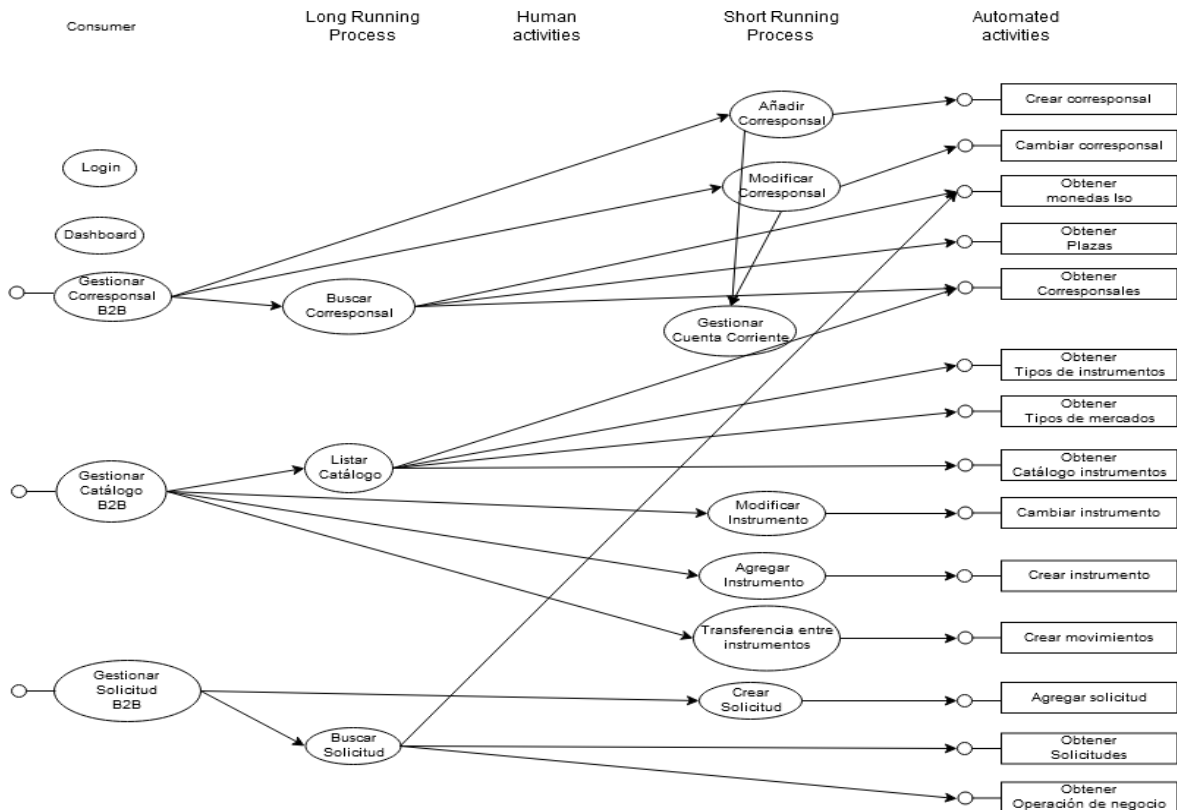


Ilustración 9 Modelo de servicio derivado de los Caso de Uso

## Modelado de servicios de objetivos

Al definir los casos de uso se identificaron aquellas actividades repetitivas y significativas del negocio de los cuales fueron identificadas y calificadas como candidatos a servicios. Los candidatos a servicios son agrupados y clasificados en el portafolio de servicios. Para el proceso se listarán los servicios candidatos y para cada servicio una lista de operaciones.

Con el objetivo de optimizar el portafolio de servicios cada servicio fue analizado en función a una serie de criterios técnicos y funcionales de los cuales se identificó los posibles servicios a ser expuestos.

Los criterios basados en las mejores prácticas para el diseño de servicios se describen a continuación: (Tomas Earl, 2008)

- **Alineación al negocio:** Evalúa el nivel de correspondencia entre el negocio y el servicio basado en los siguientes criterios:
  - Funcionalidad empresarial de soporte a los procesos y objetivos empresariales
  - Intención de financiar el servicio en todo el ciclo de vida
  - Uso y exposición del servicio a usuarios externos y/o potenciales clientes
  - Existencia actual y reutilización del servicio
- **Servicio componible:** Determina la posibilidad que el servicio participe en la composición de otros servicios de los cuales es determinado por los siguientes puntos:
  - El servicio soporta los requerimientos no funcionales
  - El servicio es sin estado
  - El servicio es independiente y puede por sí solo cumplir un requerimiento empresarial
  - El servicio puede cooperar con otros servicios para cumplir objetivos empresariales
  - La tecnología de implementación es neutral
- **Descripción:** Evalúa las capacidades del servicio de auto describirse basado en los siguientes criterios:
  - El servicio posee una descripción externa la cual distinta e independiente de la implementación física subyacente
  - El servicio puede ser descubierto y enlazado con la descripción del mismo
  - Contiene la descripción del servicio metadatos que permite la identificación del mismo

- **Reusabilidad:** Evalúa las capacidades del servicio de ser reusado por diferentes procesos del negocio
- Viabilidad técnica: Se asegura que el servicio se puede implementar y desplegar basado en los siguientes criterios:
  - El esfuerzo de implementación y gestión del servicio es razonable y fácilmente viable
  - El servicio puede ser implementado por un mínimo de sistemas proveedores
  - No existen retos tecnológicos explícitos conocidos ni anticipados

Teniendo en cuenta lo antes expuesto se obtuvo una serie de servicios que fueron considerados. En la tabla 14 se presentan los servicios y operaciones identificadas a partir del proceso.

Es importante señalar que como se ha venido reiterando se van a describir los servicios considerados más importantes para el cumplimiento de los requerimientos, toda vez que, se busca reducir el número de páginas de este trabajo y ser lo más conciso posible.

Tabla 14 Servicios candidatos

Dominio	Objeto de negocio (componente funcional)		Servicio candidato	Operaciones identificadas	
	código	nombre	Descripción	código	nombre
SWIFT	S01	msgSwift	Obtener una lista de las operaciones no identificadas por un rango	S01P01	getByRange
			Generar reporte de los montos no identificados	S01P02	generateExcel
			Obtener el mensaje SWIFT por su número	S01P03	getByFindCoMsgSwift
			Procesar los mensajes SWIFT	S01P04	processSwift
Catálogos	S02	catalog	Registrar catálogo	S02P01	insert
			Actualizar catálogo	S02P02	update
			Obtener toda la lista de catálogo	S02P03	getByFindAll
			Obtener el conjunto de valores (corresponsales, tipos de instrumentos, tipos de mercados y el catálogo de instrumentos) para la creación de un instrumento en el catálogo	S02P04	catalogComposition
			Obtener el catálogo por el código de la cuenta corriente del corresponsal	S02P05	getByFindNuCorrespCta
Instrumentos	S03	instrumento	Actualiza el estatus de un instrumento por su código único	S03P01	updateStInstrumento
			Obtener el instrumento por código unívoco	S03P02	getByFindNuInstrumento
			Obtener el instrumento por corresponsal	S03P03	getByFindNuCorrespCta
			Obtener el conjunto de valores (tipo de instrumento y el catálogo de instrumentos) para la búsqueda de instrumentos	S03P04	instrumentoComposition

			Obtener una lista de instrumentos por rango	S03P05	getByRange
			Generar reporte de los montos reales y estimados	S03P06	generateExcel
Movimientos	S09	Movimiento	Registrar un movimiento	S09P01	insert
			Obtener una lista de movimientos por un rango	S09P02	getByRange
			Obtener el conjunto de valores para la creación del movimiento	S09P03	compositionMovement
solicitudes	S13	Solicitud	Registrar una solicitud	S13P01	insert
			Establecer la hora de la ejecución de la conciliación	S13P02	startUpTimer
			Obtener el conjunto de valores (tipos de monedas, las unidades administrativas) para la creación de una nueva solicitud	S13P03	compositionNew
			Obtener el conjunto de valores (catálogo de instrumento, tipo de movimientos) para realizar la búsqueda de solicitudes	S13P04	compositionInit
			Obtener el conjunto de valores (solicitudes, tipo de movimiento) para la búsqueda de los montos no ejecutados	S13P05	compositionMNE
			Obtener un listado de solicitudes por un rango de fecha valor	S13P06	getByRange
			Hace el proceso de conciliación de las operaciones	S13P07	isReconciliation
			Obtener el conjunto de valores para la búsqueda de los montos no ejecutados	S13P08	getMontoNoEjecutado
			Generar reporte de los montos no ejecutados	S13P09	generateExcel

## Aplicar pruebas decisivas de servicio

Una vez que se hayan seleccionado y documentado los servicios candidatos en la cartera de servicios queda determinar cuáles de ellos se exponen como servicios. Teóricamente cualquier servicio candidato puede exponerse exportando su interfaz como descripción de servicio, no obstante, no todos los servicios candidatos deberían exponerse. Es decir, puede que no resulte factible hacerlo. En particular, exponer todos los métodos de todas las clases produzca un número incontenible e incontrolable de servicios, lo que llevaría al "síndrome de proliferación de servicios". (Andrea Delgado & otros, s/f)

Para evitar el síndrome de proliferación de servicios el método SOMA señala un conjunto de criterios en forma de prueba decisiva de servicio que deben usarse para filtrar las colecciones de servicios candidatos.

### Garantizar que el servicio está alineado con la empresa

La primera prueba de un servicio es sobre su alineación con la empresa. Si el servicio no es rastreado a una tarea o a un objetivo de la empresa, puede que no produzca los beneficios necesarios para la implementación de la arquitectura orientada a servicios. La tabla 15 contiene preguntas que si todas se responden de forma positiva, indicarán que el servicio está alineado con la empresa.

Tabla 15 Prueba decisiva de servicios. Alineación de los servicios con la empresa

Código servicio	¿Ofrece el servicio una funcionalidad empresarial necesaria que de soporte a procesos y objetivos empresariales?	¿Desea la empresa financiar el servicio en todo su ciclo de vida: suministro, gestión, dirección y mantenimiento?	¿Desea la empresa compartir el servicio interna o externamente con clientes o socios empresariales?	¿Existen actualmente o en el futuro oportunidades dentro de la empresa para reutilizar el servicio?
S01	SI	SI	SI	SI
S02	SI	SI	SI	SI
S03	SI	SI	SI	SI
S09	NO	SI	NO	NO
S13	SI	SI	SI	SI

### Garantizar que el servicio es componible

La componibilidad se define como un atributo que permite que el servicio participe en una composición de servicios. La siguiente tabla 16 responde a las preguntas referentes a si un servicio es componible.

Tabla 16 Prueba decisiva de servicios. Componibilidad de los servicios

Código servicio	¿Cumple el servicio con los atributos de calidad de servicio necesarios, tal como se define en los Requisitos no funcionales?	¿El servicio es sin estado?	¿Es el servicio independiente? ¿Puede el servicio desplegarse independientemente para cumplir un objetivo empresarial aunque pueda cooperar con otros servicios?	¿Es la tecnología de implementación del servicio neutral?
S01	SI	SI	SI	SI
S02	SI	SI	SI	SI
S03	SI	SI	SI	SI
S09	SI	SI	NO	SI
S13	SI	SI	SI	SI

**Garantizar que el servicio tiene descripción externa**

La descripción de servicio externalizada puede ser esa o estar generada automáticamente a través de herramientas o manualmente. La tabla siguiente responde a las preguntas relacionadas con estas pruebas decisivas de servicio.

Tabla 17 Pruebas decisivas de servicio. Descripción de servicio externalizada

Código servicio	¿Tiene el servicio una descripción de servicio externalizada que sea distinta e independiente de la implementación física subyacente?	¿Puede el servicio ser descubierto y enlazado con la descripción de servicio?	¿Contiene la descripción de servicio metadatos sobre sí misma?
S01	SI	SI	SI
S02	SI	SI	SI
S03	SI	SI	SI
S09	SI	NO	NO
S13	SI	SI	SI

## Garantizar que el servicio es reutilizable

Un servicio debe estar enfocado a su reutilización en el proceso de negocio, por ello la siguiente tabla responde a las preguntas relacionadas con este segmento.

*Tabla 18 Pruebas decisivas de servicios. Reutilización de servicios*

Código servicio	¿Puede este servicio ser utilizado por el interesado empresarial dentro de todos los procesos en los que se necesite su función?
S01	SI
S02	SI
S03	SI
S09	SI
S13	SI

## Garantizar que el servicio es técnicamente viable

La viabilidad técnica garantiza que el servicio se puede ejecutar (implementar y desplegar) realmente según los requisitos funcionales y no funcionales utilizando tecnologías disponibles. Para eso la tabla a continuación responde a si los servicios cumplen con este propósito.

Código servicio	¿Es el esfuerzo de implementación y gestión para el servicio razonable y fácilmente viable dados los requisitos o la infraestructura de la implementación?
S01	SI
S02	SI
S03	SI
S09	NO
S13	SI

## Modelado de servicios: Capa de abstracción lógica de servicios

Cuando se construyen varios tipos de servicios ellos pueden ser categorizados dependiendo de:

- El tipo de lógica que ellos encapsulan
- El alcance posible de reutilización que tiene la lógica
- Cómo esta lógica se relaciona con los dominios existentes de la empresa

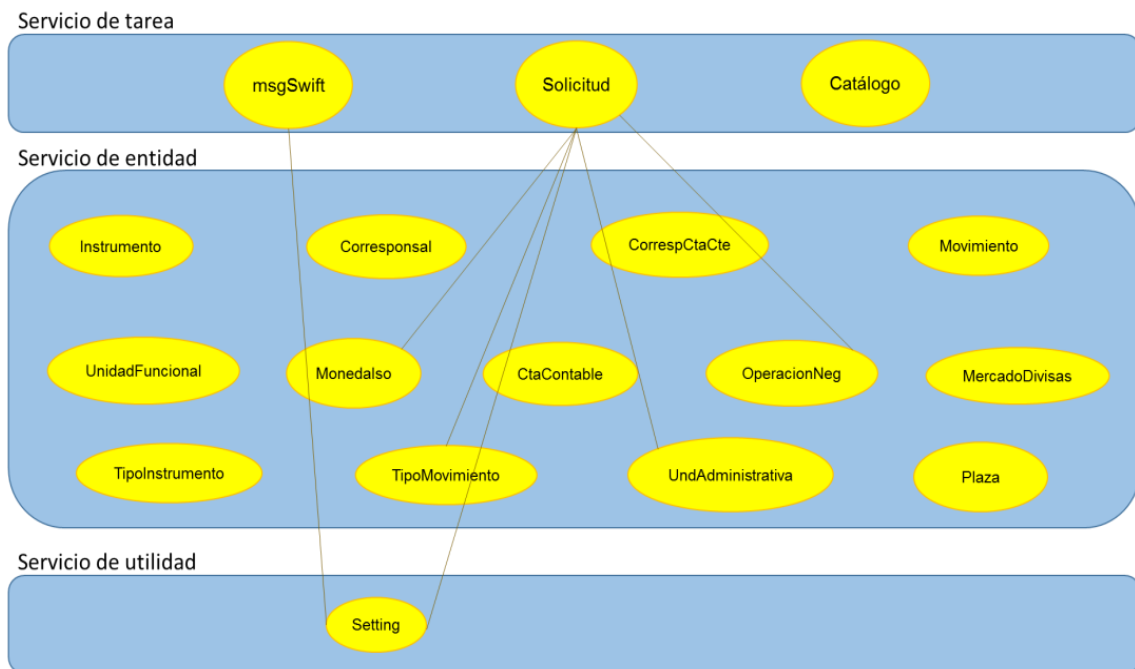


Como resultado hay tres clasificaciones comunes que representan los servicios

- Servicios de entidad
- Servicios de tarea
- Servicios de utilidad

En la siguiente figura se podrá observar cómo se categorizan los servicios candidatos obtenidos de la tabla 14.

## Modelo de servicios



*Ilustración 10 Capa de servicios*

## Identificar objetivos y subobjetivos

Los servicios que fueron identificados como candidatos tienen disposición de ser posibles a exponer, es decir, no todos los servicios identificados puede que lleguen a ser realizados. Para saber qué servicios exponer estos deben ser priorizados usando los objetivos del negocio. En ese sentido, a continuación se muestra la tabla 19 con la relación de servicios con objetivos empresariales.

Tabla 19 Matriz Procesos-Actividades-Objetivos

		Objetivos					Total prioridad
		Organización y estructuración de los instrumentos	Organización y estructuración de los corresponsales	Manejo de los flujos de fondos	Manejo de las solicitudes		
Procesos	Conciliación de operaciones del Departamento de Inversiones y Tesorería Internacional	Captación de los balances SWIFT	X	X	X		3
		Recepción de solicitudes de cobertura	X	X	X	X	3
		Generar reporte MNE			X		1
		Generar reporte MNI			X	X	1
		Remitir reportes			X		1
		Recibir operaciones adjudicadas			X	X	1
		Generar reporte balance real-estimado			X	X	1
<b>Total prioridad</b>		<b>2</b>	<b>2</b>	<b>7</b>	<b>4</b>	<b>11</b>	

Conseguida esta priorización de objetivos las mismas quedarían de la siguiente manera:

Tabla 20 Objetivos priorizados

Objetivos		Prioridad
OB01	Organización y estructuración de los instrumentos	2
OB02	Organización y estructuración de los corresponsales	2
OB03	Manejo de los flujos de fondos	7
OB04	Manejo de las solicitudes	4

Luego de tener los objetivos del negocio priorizados se asignan los servicios candidatos de la cartera de servicios del proyecto a los objetivos a los que dan soporte por lo que el siguiente paso es identificar qué servicios participan en el cumplimiento de cada objetivo del negocio. Además, también es importante establecer el nivel de importancia para el cumplimiento del mismo, para esto se definieron diferentes niveles: Mínimo (1), Medio (2), Alto (3). Siendo el valor Alto fundamental para el cumplimiento del objetivo de negocio, en cambio, un valor Mínimo tiene menos relevancia para lograr el objetivo como se ilustra en la tabla 20

Tabla 21 Identificar y asociar servicios con objetivos

			Objetivos				Total prioridad
			Organización y estructuración de los instrumentos	Organización y estructuración de los corresponsales	Manejo de los flujos de fondos	Manejo de las solicitudes de cobertura	
			OB01	OB02	OB03	OB04	
servicios	S01	MsgSwift	0	0	3	0	3
	S02	Catalog	3	3	3	3	12
	S03	Instrumento	3	1	3	3	11
	S09	Movimiento	3	3	3	3	12
	S13	Solicitud	3	1	3	3	10

Hecha la relación de los servicios con los objetivos a través de la tabla 21. Se elaboraran los servicios según las prioridades establecidas para el cumplimiento de los objetivos del proceso de negocio.

### Fase 3. Especificación de los servicios

Para esta fase corresponde la especificación de los componentes y su vinculación con los servicios de los que finalmente se expondrán junto con el diseño de los mismos y su composición a través de la orquestación de servicios. La especificación de servicios y componentes se elaboró con base a las mejores prácticas de la orientación a servicios. (Earl, T. 2008)

Entonces, los servicios identificados son diseñados, especificadas sus referencias, operaciones e interfaces.

Como se mencionó en el apartado de los Casos de Usos sólo se describiran aquí en esta fase los principales servicios para evitar extender innecesariamente el documento y no fatigar al lector.

### Arquitectura Funcional

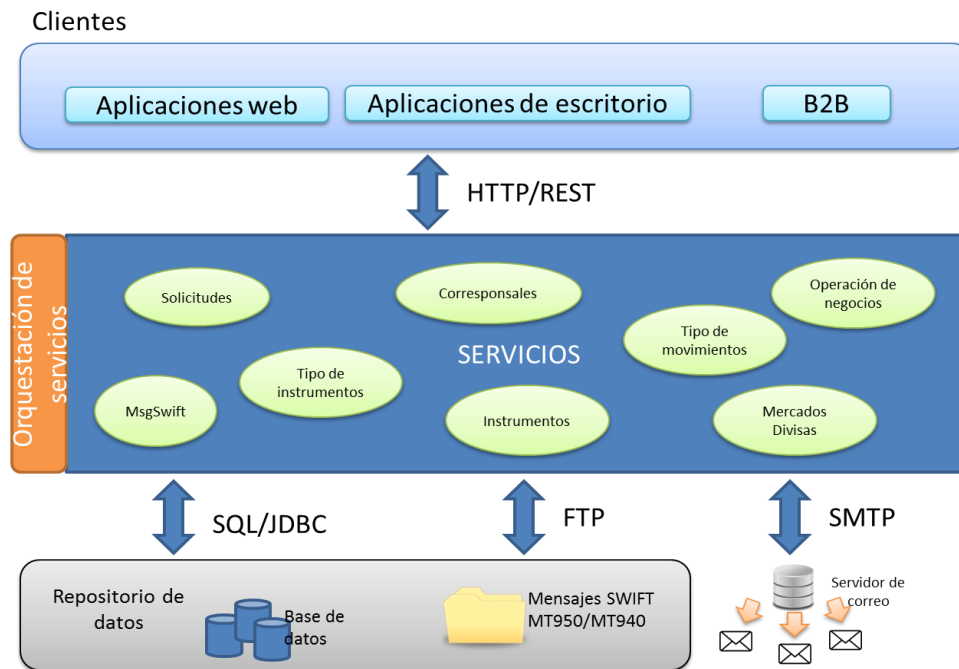
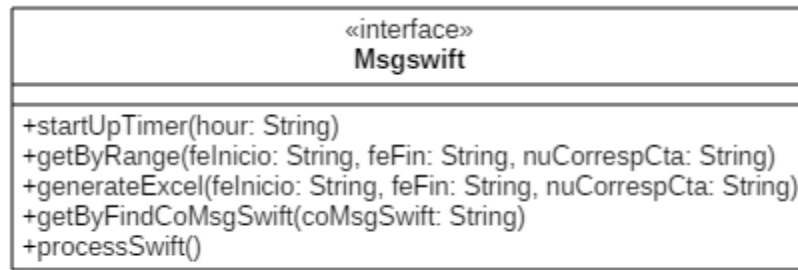


Ilustración 11 Arquitectura funcional SOA-Sistema de conciliación

- **Interfaz de servicio MsgSwift**



*Ilustración 12 Interfaz MsgSwift*

La tabla 22 a continuación describe los eventos del proceso de negocio que desencadenan las operaciones del servicio.

*Tabla 22 Eventos por operación de servicio MsgSwift*

Operación del servicio	Descripción del evento
S01P01<startUpTimer>	Permite establecer la hora de ejecución de la extracción de los balances SWIFT y conciliar las operaciones.
S01P02<getByRange>	Al ser invocado este servicio estableciendo los parámetros de búsqueda retorna una lista de los mensajes SWIFT de ese intervalo.
S01P03<generateExcel>	Retornar el reporte en Excel de las operaciones no identificadas según los parámetros suministrados.
S01P04<getByFindCoMsgSwift>	Devuelve el mensaje SWIFT por el código que lo identifica.
S01P05<processSwift>	Ejecuta la recepción y descomposición de los mensajes SWIFT MT940/MT950

### **S01P01: startUpTimer**

Esta operación permite establecer la hora de ejecución diaria de la extracción de los mensajes SWIFT MT950/MT940 recibidos y posteriormente la conciliación de las operaciones.

*Tabla 23 Estructura de mensaje de entrada S01P01*

Elemento	Descripción	Tipo de dato
hour	Es la hora en formato de 24h para establecer la hora de ejecución diaria	String

*Tabla 24 Estructura de mensaje de salida S01P01*

Elemento	Descripción	Tipo de dato
Response	Es el elemento que representa un objeto común para todos los servicios	MessageBeanXml

*Tabla 25 Estructura de excepción de S01P01*

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todas los errores que se originen en los servicios	TesoreriaServicesException

### **S01P02: getByRange**

Esta operación tiene la finalidad de obtener por un rango de fechas los mensajes SWIFT MT950/MT940 recibidos.

*Tabla 26 Estructura de mensaje de entrada S01P02*

Elemento	Descripción	Tipo de dato
felnicio	Es la fecha valor inicial del rango del que se desea obtener los mensajes SWIFT MT950/MT940	String
feffin	Es la fecha valor final del rango del que se desea obtener los mensajes SWIFT MT950/940	String
nuCorrespCta	Es el número de la cuenta corriente con el que se puede buscar un mensaje SWIFT.	String

*Tabla 27 Estructura de mensaje de salida S01P02*

Elemento	Descripción	Tipo de dato
Response	Es el elemento que representa un objeto común para todos los servicios.	MessageBeanXml

*Tabla 28 Estructura de excepción de S01P02*

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todas los errores que se originen en los servicios.	TesoreriaServicesException

### **S01P03: generateExcel**

Tiene como finalidad generar el reporte en Excel de los montos no identificados en un rango de fecha valor de las operaciones que están en los balances SWIFT

*Tabla 29 Estructura de mensaje de entrada S01P03*

Elemento	Descripción	Tipo de dato
felnicio	Es la fecha valor inicial del rango del que se desea obtener las operaciones no identificados para generar el reporte	String
feфин	Es la fecha valor final del rango del que se desea obtener las operaciones no identificadas para generar el reporte	String
nuCorrespCta	Es el número de la cuenta corriente con el que se puede buscar un mensaje SWIFT.	String

*Tabla 30 Estructura de mensaje de salida S01P03*

Elemento	Descripción	Tipo de dato
Response	Es un objeto que contiene el archivo Excel representado en octet-stream	ResponseBuilder

*Tabla 31 Estructura de excepción de S01P03*

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todas los errores que se originen en los servicios	TesoreriaServicesException

### **S01P04: getByFindCoMsgSwift**

Expone la búsqueda de un mensaje SWIFT a través de su código de identificación único devolviendo, en caso de existir, todo el mensaje SWIFT con sus datos respectivos.

*Tabla 32 Estructura de mensaje de entrada S01P04*

Elemento	Descripción	Tipo de dato
coMsgSwift	Es el código inequívoco para la búsqueda del mensaje SWIFT	String

*Tabla 33 Estructura de mensaje de salida S01P04*

Elemento	Descripción	Tipo de dato
Response	Contiene los datos del mensaje SWIFT	MsgSwiftBeanXml

*Tabla 34 Estructura de excepción de S01P04*

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todas los errores que se originen en los servicios	TesoreriaServicesException

## S01P05: processSwift

Realiza la búsqueda y descomposición de los mensajes SWIFT emitidos por los corresponsales almacenando los datos obtenidos

Tabla 35 Estructura de mensaje de salida S01P05

Elemento	Descripción	Tipo de dato
Response	Devuelve un mensaje de texto indicando lo satisfactorio del proceso	MessageBeanXml

Tabla 36 Estructura de excepción de S01P05

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todas los errores que se originen en los servicios	TesoreriaServicesException

- **Interfaz de servicio Catalogo**

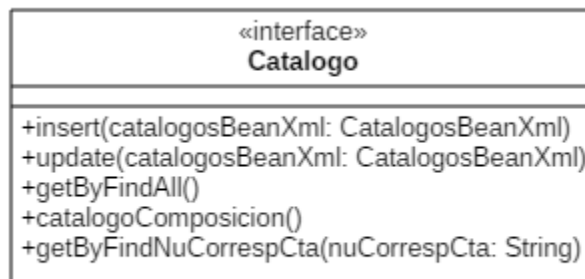


Ilustración 13 Interfaz de servicio Catalogo

La tabla 37 describe los eventos del proceso de negocio que desencadenan las operaciones del servicio.

Tabla 37 Eventos por operación de servicio Catalogo

Operación del servicio	Descripción del evento
S02P01<insert>	Almacena el catálogo con sus instrumentos generando los movimientos de la operación de la apertura de saldos de esos instrumentos
S02P02<update>	Modifica el catálogo y sus instrumentos generando los movimientos de la operación de la modificación, de ser el caso, de los saldos de esos instrumentos
S02P03<getByFindAll>	Devuelve toda la lista del catálogo
S02P04<catalogoComposicion>	Invoca los servicios que proveen la lista de



	corresponsales por estatus, la lista de los tipos de instrumentos, la lista de los tipos de mercado de divisas y ejecuta la operación de la búsqueda de catálogos
S02P05<getByFindNuCorrespCta>	Obtiene la lista de los catálogos por el código de la cuenta corriente del corresponsal

### S02P01: insert

Realiza el almacenamiento de un nuevo catálogo con sus instrumentos asociados generando los movimientos de la operación de apertura de saldos de esos instrumentos

*Tabla 38 Estructura de mensaje de entrada S02P01*

Elemento	Descripción	Tipo de dato
catalogosBeanXml	El objeto compuesto por los atributos que componen el catálogo y los atributos del instrumento	CatalogosBeanXml

*Tabla 39 Estructura de mensaje de salida S02P01*

Elemento	Descripción	Tipo de dato
Response	Devuelve los datos del instrumento creado	InstrumentoBeanXml

*Tabla 40 Estructura de excepción de S02P01*

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todas los errores que se originen en los servicios	TesoreriaServicesException

### S02P02: update

Es el encargado de actualizar los datos del catálogo

*Tabla 41 Estructura de mensaje de entrada S02P02*

Elemento	Descripción	Tipo de dato
catalogBeanXml	El objeto compuesto por los atributos que componen el catálogo	CatalogBeanXml

*Tabla 42 Estructura de mensaje de salida S02P02*

Elemento	Descripción	Tipo de dato
Response	Devuelve únicamente el código de estado de haber invocado la operación	ResponseBuilder

Tabla 43 Estructura de excepción de S02P02

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todas los errores que se originen en los servicios	TesoreriaServicesException

### S02P03: getByFindAll

Devuelve la lista del catálogo completo

Tabla 44 Estructura de mensaje de entrada S02P03

Elemento	Descripción	Tipo de dato
Request	El objeto formado por el http request	HttpHeaders

Tabla 45 Estructura de mensaje de salida S02P03

Elemento	Descripción	Tipo de dato
reporte	Devuelve la lista completa del catálogo	File

Tabla 46 Estructura de excepción de S02P03

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todas los errores que se originen en los servicios	TesoreriaServicesException

### S02P04: CatalogoComposicion

Es un servicio compuesto que se encarga de invocar el servicio de corresponsal que devuelve la lista de los corresponsales, el servicio de tipo de instrumento que devuelve la lista de los tipos de instrumentos, el servicio de mercado divisas que devuelve la lista de tipos de mercados, y llama a la operación de consultar el listado de los catálogos.

Tabla 47 Estructura de mensaje de salida S02P04

Elemento	Descripción	Tipo de dato
Response	Devuelve el objeto que contiene toda la listas de los corresponsales, los tipos de instrumentos, los tipos de mercados y la lista de catálogo	CatalogoCompositionBeanXml

Tabla 48 Estructura de mensaje de excepción de S02P04

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todas los errores que se originen en los servicios	TesoreriaServicesException

## S02P05: getByFindNuCorrespCta

Al ser invocado el servicio hace el llamado a la operación de búsqueda del listado del catálogo por el código inequívoco de la cuenta corriente del corresponsal

Tabla 49 Estructura de mensaje de entrada S02P05

Elemento	Descripción	Tipo de dato
nuCorrespCta	El código univoco de la cuenta corriente por el cual se va a realizar la búsqueda de los catálogos	String

Tabla 50 Estructura de mensaje de salida S02P05

Elemento	Descripción	Tipo de dato
CatalogoBeanXml	Devuelve la lista del catálogo encontrado por el código único de la cuenta corriente del corresponsal	CatalogoBeanXml

Tabla 51 Estructura de mensaje de excepción de S02P05

Elemento	Descripción	Tipo de dato
Exception	De no conseguir la lista del catálogo por el código suministrado devuelve un mensaje indicando tal eventualidad  Es el elemento que representa un objeto común para todas los errores que se originen en los servicios	TesoreriaServicesException

- **Interfaz del servicio de instrumento**

«interface» Instrumento
<pre> +insert(instrumentoBeanXml: InstrumentoBeanXml) +insert(movimientosBeanXml: MovimientosBeanXml) +updateStInstrumento(nulInstrumento: short) +getByFindNuInstrumento(nulInstrumento: short) +getByFindNuCorrespCta(nuCorrespCta: String) +instrumentoComposition() +getByRange(feInicio: String, feFin: String, nulInstrumento: short, coTipolnstr: String) +generateExcel(feInicio: String, feFin: String, nulInstrumento: short, coTipolnstr: String) +getMovimientos(feInicio: String, feFin: String, nulInstrumento: Short) </pre>

Ilustración 14 Interfaz del servicio de instrumento

La tabla 52 describe los eventos del proceso de negocio que desencadenan las operaciones del servicio.

Tabla 52 Eventos por operación de servicio Instrumento

Operación del servicio	Descripción del evento
S03P01<insert>	Guarda los datos de un nuevo instrumento

S03P02<insert>	Ingresa los datos del movimiento originado por una transferencia entre instrumentos
S03P03<updateStInstrumento>	Actualiza el estatus del instrumento a inactivo/activo según el código unívoco suministrado
S03P04<getByFindNuInstrumento>	Obtiene el conjunto de valores (cuentas corrientes, plaza, tipo de monedas, cuentas contables y el instrumento) que conforman el instrumento.
S03P05<getByFindNuCorrespCta>	Obtiene una lista de instrumentos que contengan la relación con el código único de la cuenta corriente del corresponsal
S03P06<instrumentoComposition>	Obtiene la lista de catálogos y los tipos de instrumentos para la búsqueda de instrumentos
S03P07<getByRange>	Obtiene una lista de instrumentos según el rango de la fecha valor suministrado
S03P08<generateExcel>	Devuelve un archivo Excel de los montos reales de los instrumentos según los parámetros suministrados
S03P09<getMovimientos>	Obtiene los movimientos a través de los parámetros de búsqueda

### S03P01: insert

Almacena los datos que conforman la creación de un nuevo instrumento.

*Tabla 53 Estructura de mensaje de entrada S03P01*

Elemento	Descripción	Tipo de dato
instrumentoBeanXml	El objeto contentivo de las propiedades que tienen los datos de un instrumento.	InstrumentoBeanXml

*Tabla 54 Estructura de mensaje de salida S03P01*

Elemento	Descripción	Tipo de dato
Response	Devuelve los datos del instrumento creado.	InstrumentoBeanXml

*Tabla 55 Estructura de excepción de S03P01*

Elemento	Descripción	Tipo de dato
Exception	De no haber suministrado los datos completos a través del objeto se lanza un error indicando el error ocurrido.  Es el elemento que representa un objeto común para todos los errores que se originen en los servicios.	TesoreriaServicesException

### S03P02: insert

Ingresa los datos del movimiento originado por el traspaso de fondos entre instrumentos.

*Tabla 56 Estructura de mensaje de entrada S03P02*

Elemento	Descripción	Tipo de dato
MovimientosBeanXml	El objeto contentivo de los datos de la operación originada por la transferencia de fondos entre instrumentos.	MovimientosBeanXml

*Tabla 57 Estructura de mensaje de salida S03P02*

Elemento	Descripción	Tipo de dato
Response	Devuelve los datos del instrumento creado.	InstrumentoBeanXml

*Tabla 58 Estructura de excepción de S03P02*

Elemento	Descripción	Tipo de dato
Exception	De no haber suministrado los datos completos a través del objeto se lanza un error indicando el error ocurrido. Es el elemento que representa un objeto común para todos los errores que se originen en los servicios.	TesoreriaServicesException

### S03P03: updateStInstrumento

Actualiza el estatus del instrumento según el código del instrumento suministrado.

*Tabla 59 Estructura de mensaje de entrada S03P03*

Elemento	Descripción	Tipo de dato
nulInstrumento	El código unívoco del instrumento al que se desea cambiar el estatus	Short

*Tabla 60 Estructura de mensaje de salida S03P03*

Elemento	Descripción	Tipo de dato
Response	Devuelve los datos del instrumento modificado	InstrumentoBeanXml

*Tabla 61 Estructura de excepción de S03P03*

Elemento	Descripción	Tipo de dato
Exception	De no existir el código suministrado se lanza un error indicando el error ocurrido Es el elemento que representa un objeto común	TesoreriaServicesException

	para todas los errores que se originen en los servicios	
--	---	--

### S03P04: getByFindNuInstrumento

Devuelve el conjunto de valores que conforman el instrumento. (cuentas corrientes, plazas, tipos de monedas, cuentas contables y el instrumento)

*Tabla 62 Estructura de mensaje de entrada S03P04*

Elemento	Descripción	Tipo de dato
nuInstrumento	El código unívoco del instrumento al que se desea obtener el conjunto de valores que son parte del instrumento	Short

*Tabla 63 Estructura de mensaje de salida S03P04*

Elemento	Descripción	Tipo de dato
Response	Devuelve la lista de las cuentas corrientes, la lista de las plazas, la lista de las monedas, la lista de las cuentas contables y el instrumento	InstrumentoCompositionBeanXml

*Tabla 64 Estructura de excepción de S03P04*

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todas los errores que se originen en los servicios	TesoreriaServicesException

### S03P05: getByFindNuCorrespCta

Retorna la lista de instrumentos asociados a la cuenta corriente suministrada

*Tabla 65 Estructura de mensaje de entrada S03P05*

Elemento	Descripción	Tipo de dato
nuCorrespCta	El código unívoco de la cuenta corriente al que se desea buscar la lista de instrumentos	String

*Tabla 66 Estructura de mensaje de salida S03P05*

Elemento	Descripción	Tipo de dato
Response	Devuelve la lista de las cuentas corrientes, la lista de las plazas, la lista de las monedas, la lista de las cuentas contables y el instrumento	InstrumentosBeanXml

Tabla 67 Estructura de excepción de S03P05

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todas los errores que se originen en los servicios	TesoreriaServicesException

### S03P06: instrumentoComposition

Se obtiene el conjunto de valores por el que se puede hacer una búsqueda de instrumento

Tabla 68 Estructura de mensaje de salida S03P06

Elemento	Descripción	Tipo de dato
Response	Devuelve la lista de tipos de instrumentos y la lista de catálogos que se pueden usar para la búsqueda de un instrumento	InstrumentoCompositionBeanXml

Tabla 69 Estructura de excepción de S03P06

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todas los errores que se originen en los servicios	TesoreriaServicesException

### S03P07: getByRange

Se obtiene una lista de instrumentos por el rango de fecha valor suministrado

Tabla 70 Estructura de mensaje de entrada S03P07

Elemento	Descripción	Tipo de dato
feInicio	La fecha de inicio a partir de donde se desea obtener el listado de instrumentos	String
feFin	La fecha final hasta donde se desea obtener el listado de instrumentos	String

Tabla 71 Estructura de mensaje de salida S03P07

Elemento	Descripción	Tipo de dato
Response	Devuelve la lista de instrumentos según el rango de fechas valor indicados	InstrumentosBeanXml

Tabla 72 Estructura de excepción de S03P07

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todas los errores que se originen en los servicios	TesoreriaServicesException

### S03P08: generateExcel

Devuelve el reporte en Excel de los montos reales (la posición del instrumento) según un rango de fecha valor

*Tabla 73 Estructura de mensaje de entrada S03P08*

Elemento	Descripción	Tipo de dato
feInicio	La fecha de inicio a partir de donde se desea obtener el listado de la posición del instrumento	String
feFin	La fecha final hasta donde se desea obtener el listado de la posición del instrumento	String

*Tabla 74 Estructura de mensaje de salida S03P08*

Elemento	Descripción	Tipo de dato
reporte	Devuelve la lista de instrumentos según el rango de fechas valor indicados	File

*Tabla 75 Estructura de excepción de S03P08*

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todas los errores que se originen en los servicios	TesoreriaServicesException

### S03P09: getMovimientos

Devuelve los movimientos generados por las transacciones de los instrumentos.

*Tabla 76 Estructura de mensaje de entrada S03P09*

Elemento	Descripción	Tipo de dato
feInicio	La fecha de inicio a partir de donde se desea obtener el listado de la posición del instrumento	String
feFin	La fecha final hasta donde se desea obtener el listado de la posición del instrumento	String

*Tabla 77 Estructura de mensaje de salida S03P09*

Elemento	Descripción	Tipo de dato
reporte	Devuelve la lista de instrumentos según el rango de fechas valor indicados	File



Tabla 78 Estructura de excepción de S03P09

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todas los errores que se originen en los servicios	TesoreriaServicesException

- **Interfaz del servicio de Solicitud**

«interface» Solicitud
<pre> +insert(solicitudBeanXml: SolicitudBeanXml) +startUpTimer(hour: string) +compositionNew() +compositionInit() +compositionMNE() +getByRange(feInicio: String, feFin: String, nulInstrumento: short, stSolicitud: String) +isReconciliation(feInicio: String, feFin: String) +getMontoNoEjecutado(feInicio: String, feFin: String, nulInstrumento: short) +generateExcel(feInicio: String, feFin: String, nulInstrumento: short)                     </pre>

Ilustración 15 Interfaz del servicio de Solicitud

La tabla 79 describe los eventos del proceso de negocio que desencadenan las operaciones del servicio.

Tabla 79 Eventos por operación de servicio Solicitud

Operación del servicio	Descripción del evento
S13P01<insert>	Guarda los datos una nueva solicitud de cobertura generada por los departamentos del BCV.
S13P02<startUpTime>	Establece la hora de ejecución automática para el proceso de conciliación de las operaciones.
S13P03<compositionNew>	Obtiene el listado de los departamentos del BCV y el listado de las monedas ISO habilitadas para la creación de una nueva solicitud de cobertura.
S13P04<compositionInit>	Devuelve el listado del catálogo de instrumentos y el listado de los tipos de movimientos para hacer la búsqueda de solicitudes de cobertura.
S13P05<compositionMNE>	Obtiene el listado de las operaciones no ejecutadas y el listado de los tipos de movimiento para la búsqueda en el reporte MNE.
S13P06<getByRange>	Devuelve el resultado de la búsqueda de las solicitudes según los parámetros.
S13P07<isReconciliation>	Permite ejecutar el proceso de conciliación manualmente.

*Tabla 80 Eventos por operación de servicio Solicitud*

Operación del servicio	Descripción del evento
S13P08<getMontoNoEjecutado>	Devuelve el listado de las operaciones que no fueron ejecutadas según los parámetros de búsqueda.
S13P09<generateExcel>	Genera el reporte en Excel de los montos no ejecutados según los parámetros recibidos.

### **S13P01: insert**

Almacena los datos de la solicitud de cobertura.

*Tabla 81 Estructura de mensaje de entrada S13P01*

Elemento	Descripción	Tipo de dato
solicitudBeanXml	El objeto con las propiedades para almacenar la solicitud de cobertura generada por los departamentos del BCV.	SolicitudBeanXml

*Tabla 82 Estructura de mensaje de salida S13P01*

Elemento	Descripción	Tipo de dato
messageBeanXml	Devuelve el mensaje de texto de la creación de la creación de la solicitud.	MessageBeanXml

*Tabla 83 Estructura de excepción de S13P01*

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todos los errores que se originen en los servicios.	TesoreriaServicesException

### **S13P02: startUpTime**

Establece la hora de ejecución automática para la conciliación de las operaciones generadas en los días hábiles por los departamentos del BCV.

*Tabla 84 Estructura de mensaje de entrada S13P02*

Elemento	Descripción	Tipo de dato
hour	La hora en formato de 24h.	String

*Tabla 85 Estructura de mensaje de salida S13P02*

Elemento	Descripción	Tipo de dato
messageBeanXml	Devuelve el mensaje de texto de la hora establecida para la conciliación.	MessageBeanXml

Tabla 86 Estructura de excepción de S13P02

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todos los errores que se originen en los servicios.	TesoreriaServicesException

### S13P03: compositionNew

Devuelve el conjunto de valores (lista de monedas ISO y lista de las unidades administrativas) para la creación de una nueva solicitud de cobertura.

Tabla 87 Estructura de mensaje de salida S13P03

Elemento	Descripción	Tipo de dato
solicitudCompositionBeanXml	Devuelve el listado de las monedas Iso y el listado de las unidades administrativas para la creación de una nueva solicitud de cobertura.	SolicitudCompositionBeanXml

Tabla 88 Estructura de excepción de S13P03

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todos los errores que se originen en los servicios.	TesoreriaServicesException

### S13P04: compositionInit

Devuelve el conjunto de valores (lista del catálogo de los instrumentos y la lista de los tipos de movimientos) para realizar la búsqueda de solicitudes.

Tabla 89 Estructura de mensaje de salida S13P04

Elemento	Descripción	Tipo de dato
solicitudCompositionInitBeanXml	Devuelve el objeto contentivo con el listado de los instrumentos y el listado de los tipos de movimientos.	SolicitudCompositionInitBeanXml

Tabla 90 Estructura de excepción de S13P04

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todos los errores que se originen en los servicios.	TesoreriaServicesException

### S13P05: compositionMNE

Retorna el listado de las operaciones no ejecutadas y el listado de los tipos de movimientos para la búsqueda de los montos no ejecutados.

Tabla 91 Estructura de mensaje de salida S13P05

Elemento	Descripción	Tipo de dato
solicitudCompositionInitBeanXml	Devuelve el conjunto de valores (lista de operaciones no ejecutadas y lista de tipos de movimientos).	SolicitudCompositionInitBeanXml

Tabla 92 Estructura de excepción de S13P05

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todos los errores que se originen en los servicios.	TesoreriaServicesException

### S13P06: getByRange

Retorna el listado de solicitudes según los parámetros de entrada para su búsqueda.

Tabla 93 Estructura de mensaje de entrada S13P06

Elemento	Descripción	Tipo de dato
feInicio	La fecha valor inicial con la que se va a realizar la búsqueda de las solicitudes.	String
feFin	La fecha valor final hasta donde se hará la búsqueda de las solicitudes.	String
nulInstrumento	Código del instrumento al que se va a asociar las solicitudes de cobertura para su búsqueda.	short
stSolicitud	Estatus de la solicitud de cobertura con el que se desea obtener el listado de las solicitudes.	String

Tabla 94 Estructura de mensaje de salida S13P06

Elemento	Descripción	Tipo de dato
solicitudesBeanXml	Devuelve el listado de las solicitudes encontrada según los parámetros de búsqueda indicados.	SolicitudesBeanXml

Tabla 95 Estructura de excepción de S13P06

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todos los errores que se originen en los servicios.	TesoreriaServicesException

### S13P07: isReconciliation

Permite realizar la ejecución manual de la conciliación de las operaciones según la fecha valor establecido.

Tabla 96 Estructura de mensaje de entrada S13P07

Elemento	Descripción	Tipo de dato
feInicio	La fecha valor inicial con la que se va a realizar la conciliación de las operaciones.	String
feFin	La fecha valor final hasta donde se hará la conciliación de las operaciones.	String

Tabla 97 Estructura de mensaje de salida S13P07

Elemento	Descripción	Tipo de dato
messageBeanXml	Devuelve el un mensaje de texto indicado el proceso satisfactorio de la conciliación.	MessageBeanXml

Tabla 98 Estructura de excepción de S13P07

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todos los errores que se originen en los servicios.	TesoreriaServicesException

### S13P08: getMontoNoEjecutado

Retorna el listado de las operaciones que no fueron ejecutadas según los parámetros de búsqueda indicados.

Tabla 99 Estructura de mensaje de entrada S13P08

Elemento	Descripción	Tipo de dato
feInicio	La fecha valor inicial con la que se va a realizar la conciliación de las operaciones.	String
feFin	La fecha valor final hasta donde se hará la conciliación de las operaciones.	String

nulInstrumento	Código de instrumento para realizar la búsqueda de los montos no ejecutados.	short
----------------	--	-------

*Tabla 100 Estructura de mensaje de salida S13P08*

Elemento	Descripción	Tipo de dato
solicitudesBeanXml	Devuelve el listado de los montos que no fueron ejecutados con base a los parámetros indicados.	SolicitudesBeanXml

*Tabla 101 Estructura de excepción de S13P08*

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todos los errores que se originen en los servicios.	TesoreriaServicesException

### **S13P09: generateExcel**

Retorna el reporte en Excel de las operaciones que no fueron ejecutadas con base a los parámetros de búsqueda.

*Tabla 102 Estructura de mensaje de entrada S13P09*

Elemento	Descripción	Tipo de dato
feInicio	La fecha valor inicial con la que se va a realizar el reporte de las operaciones.	String
feFin	La fecha valor final hasta donde se hará el reporte de las operaciones.	String
nulInstrumento	Código de instrumento para realizar el reporte de los montos no ejecutados.	short

*Tabla 103 Estructura de mensaje de salida S13P09*

Elemento	Descripción	Tipo de dato
Response	Devuelve el archivo Excel del reporte generado.	ResponseBuilder

*Tabla 104 Estructura de excepción de S13P09*

Elemento	Descripción	Tipo de dato
Exception	Es el elemento que representa un objeto común para todos los errores que se originen en los servicios.	TesoreriaServicesException

## Vista de comportamiento

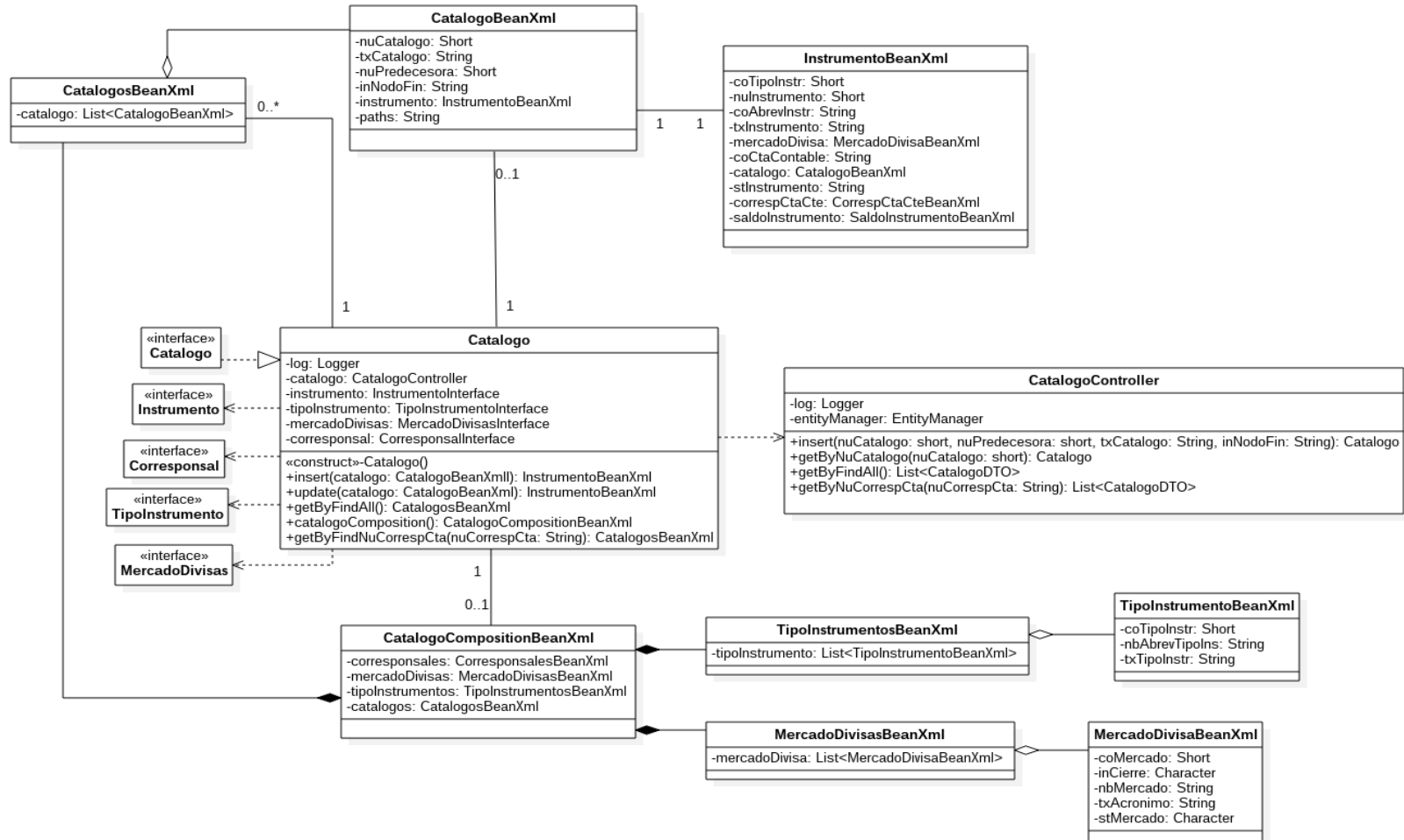


Ilustración 16 Clases por servicios. Catálogo

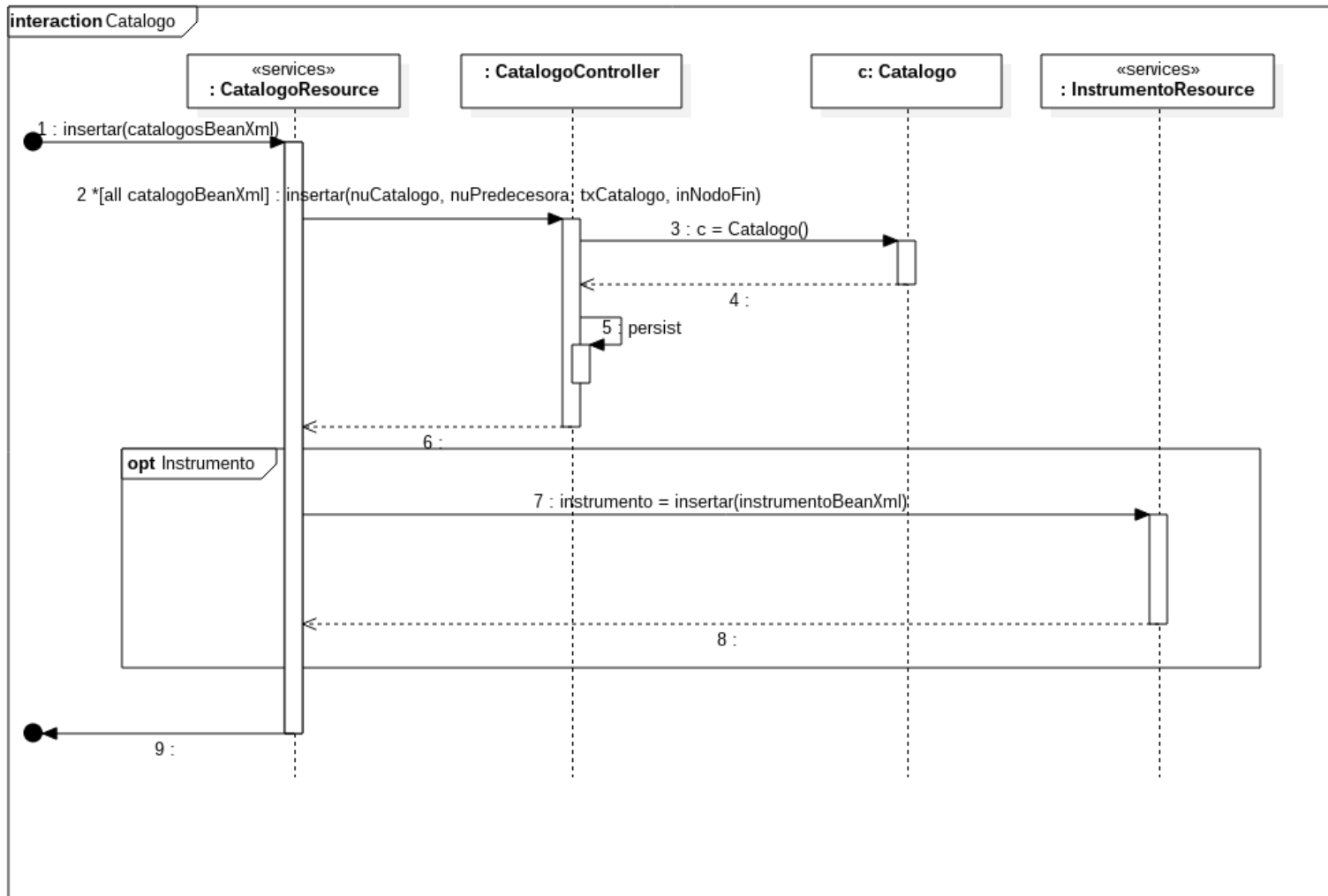


Ilustración 17 Secuencia por servicios. Catálogo



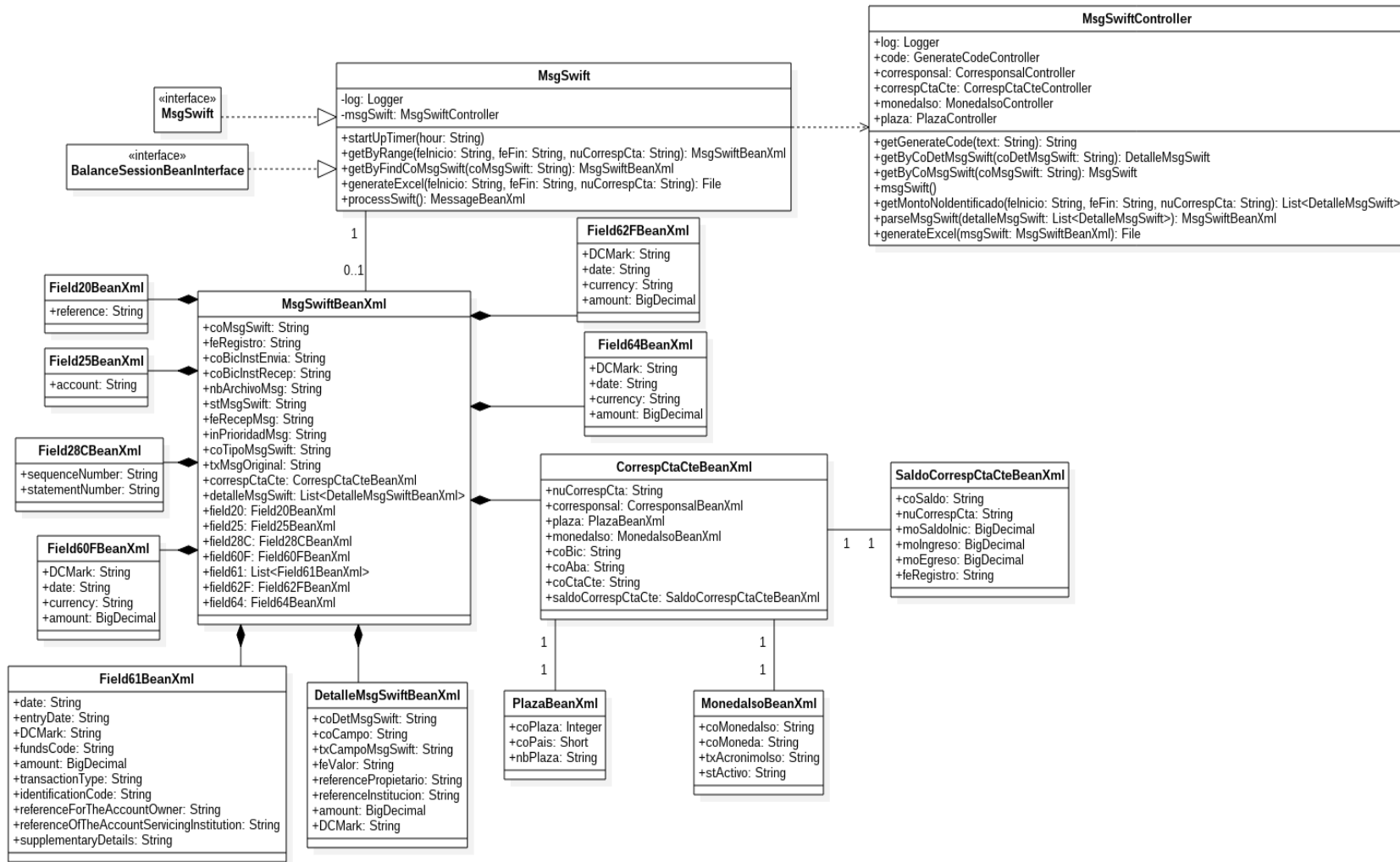


Ilustración 18 Clases por servicios. MsgSwift

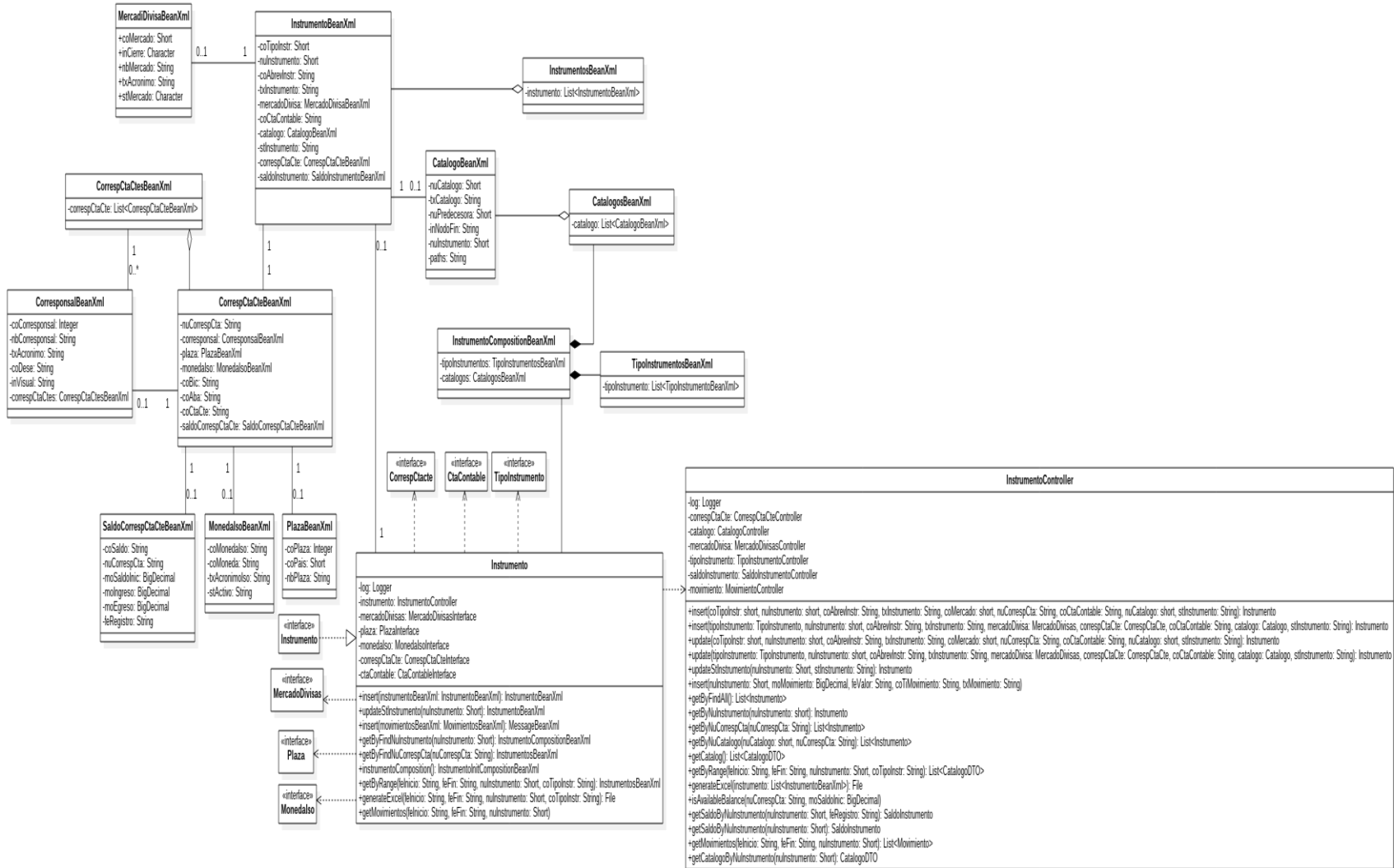


Ilustración 19 Clases por servicios. Instrumento

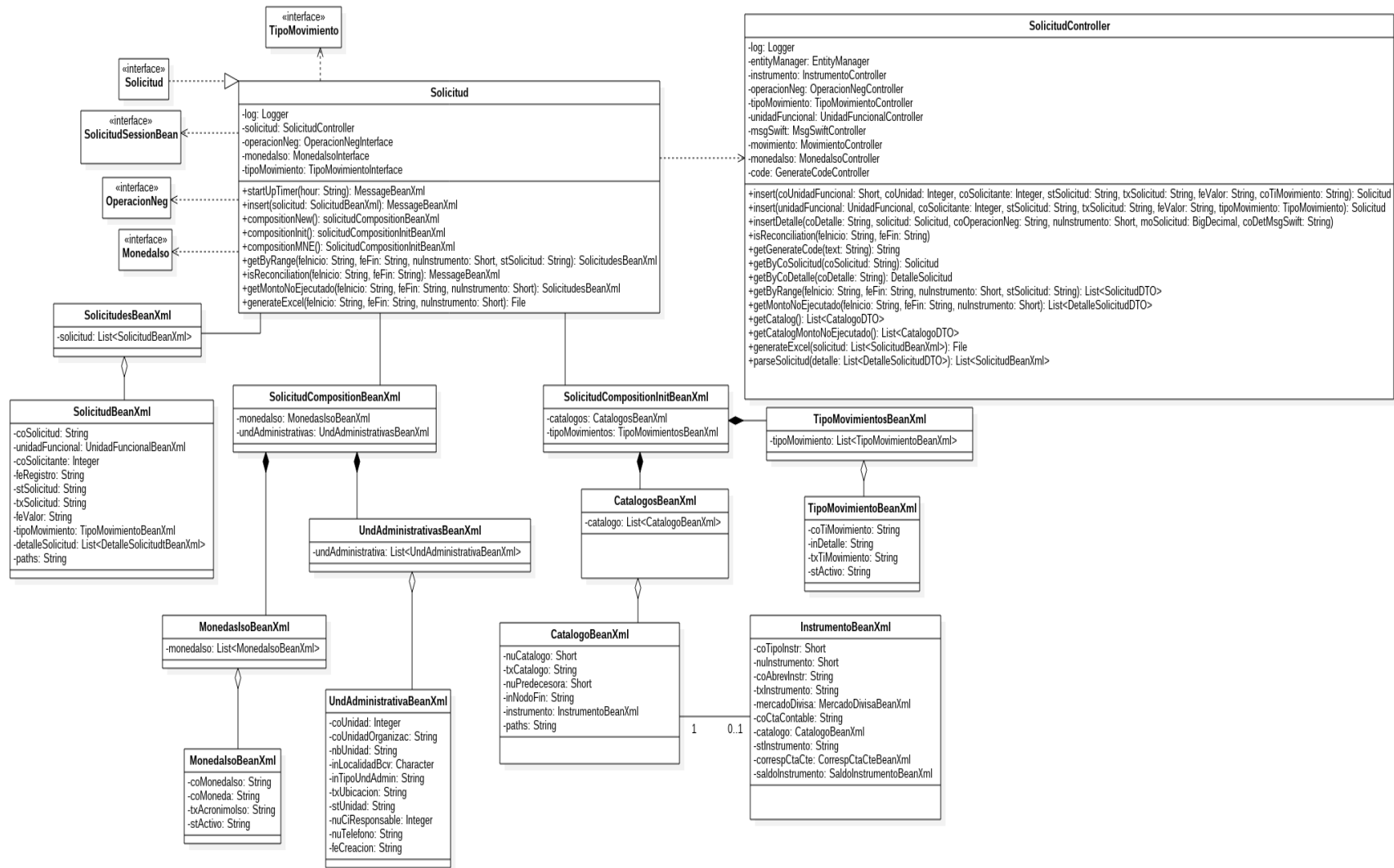


Ilustración 20 Clases por servicios. Solicitud

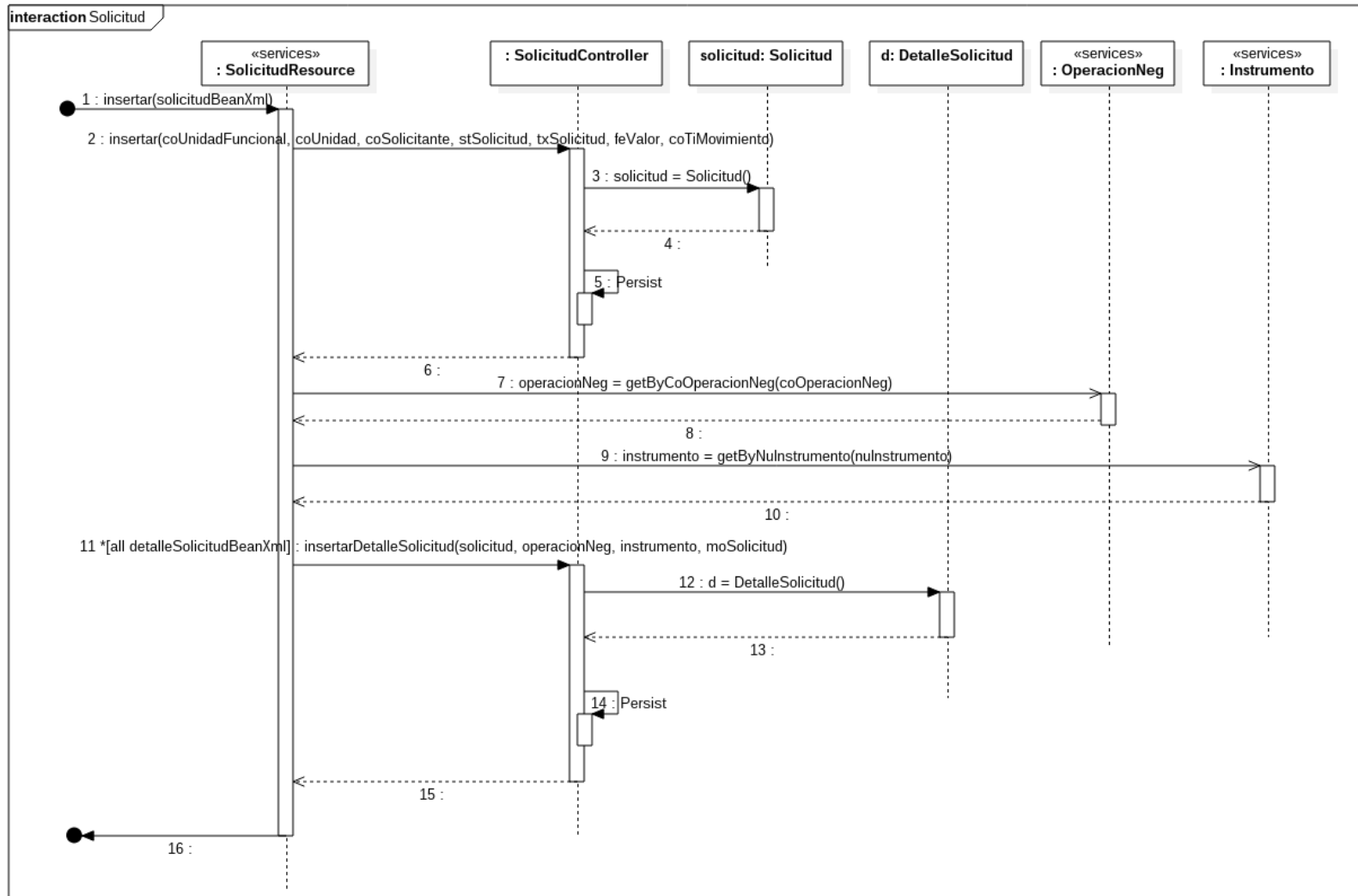


Ilustración 21 Secuencia por servicio. Solicitud

## Modelado de componentes de servicios

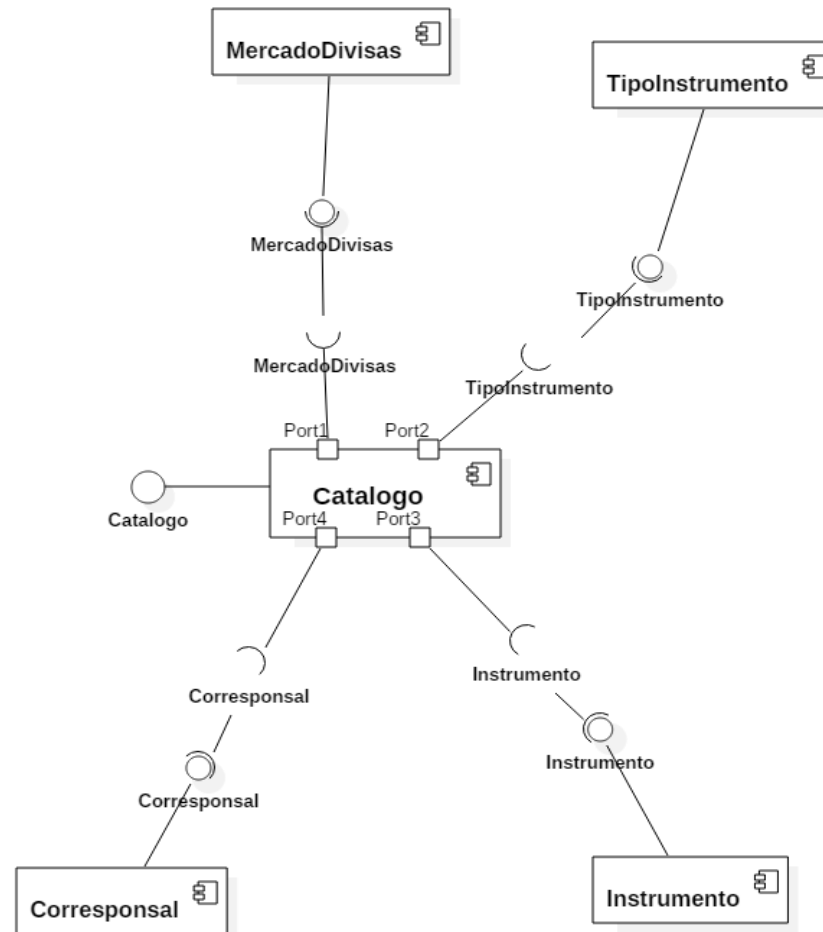
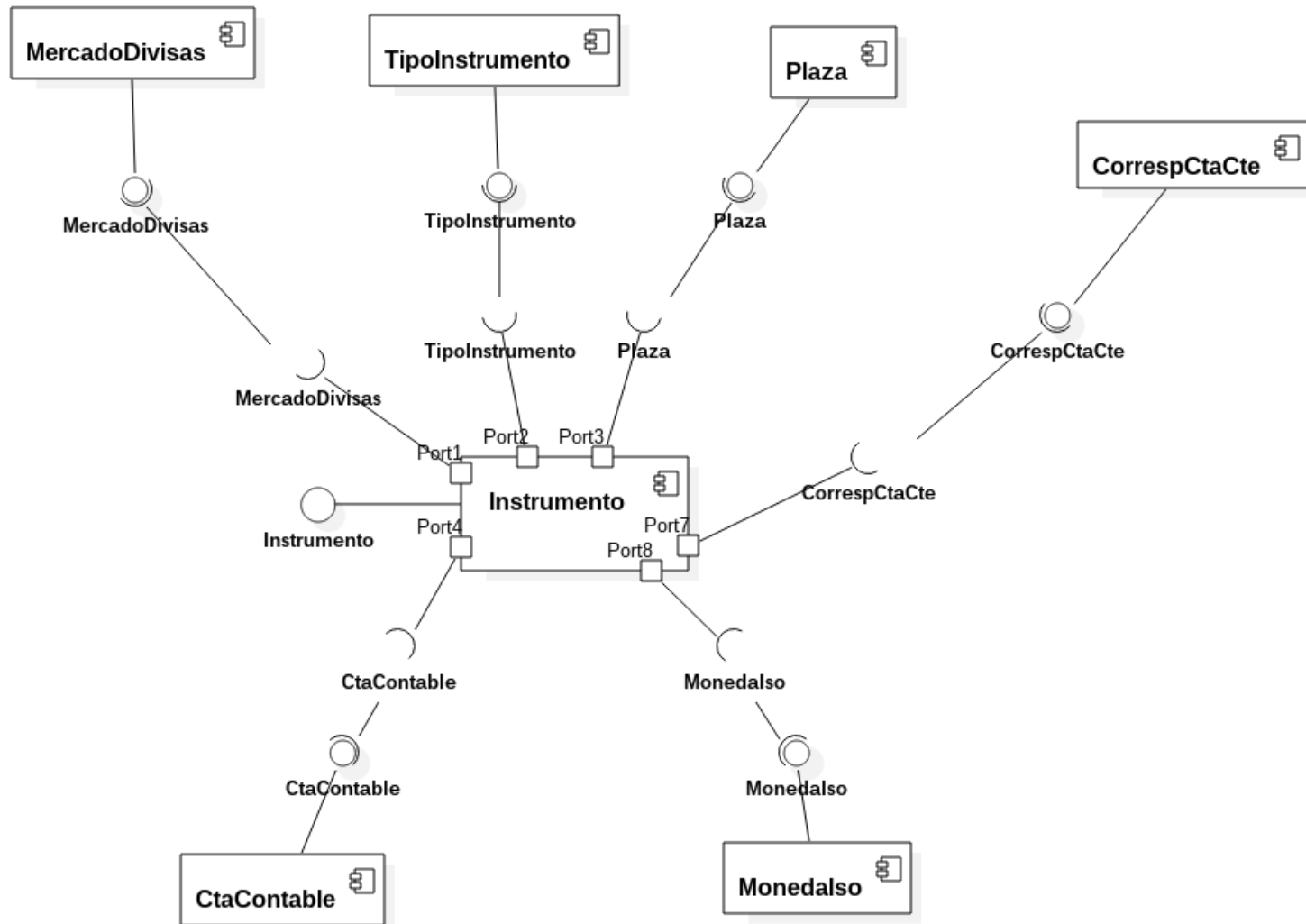
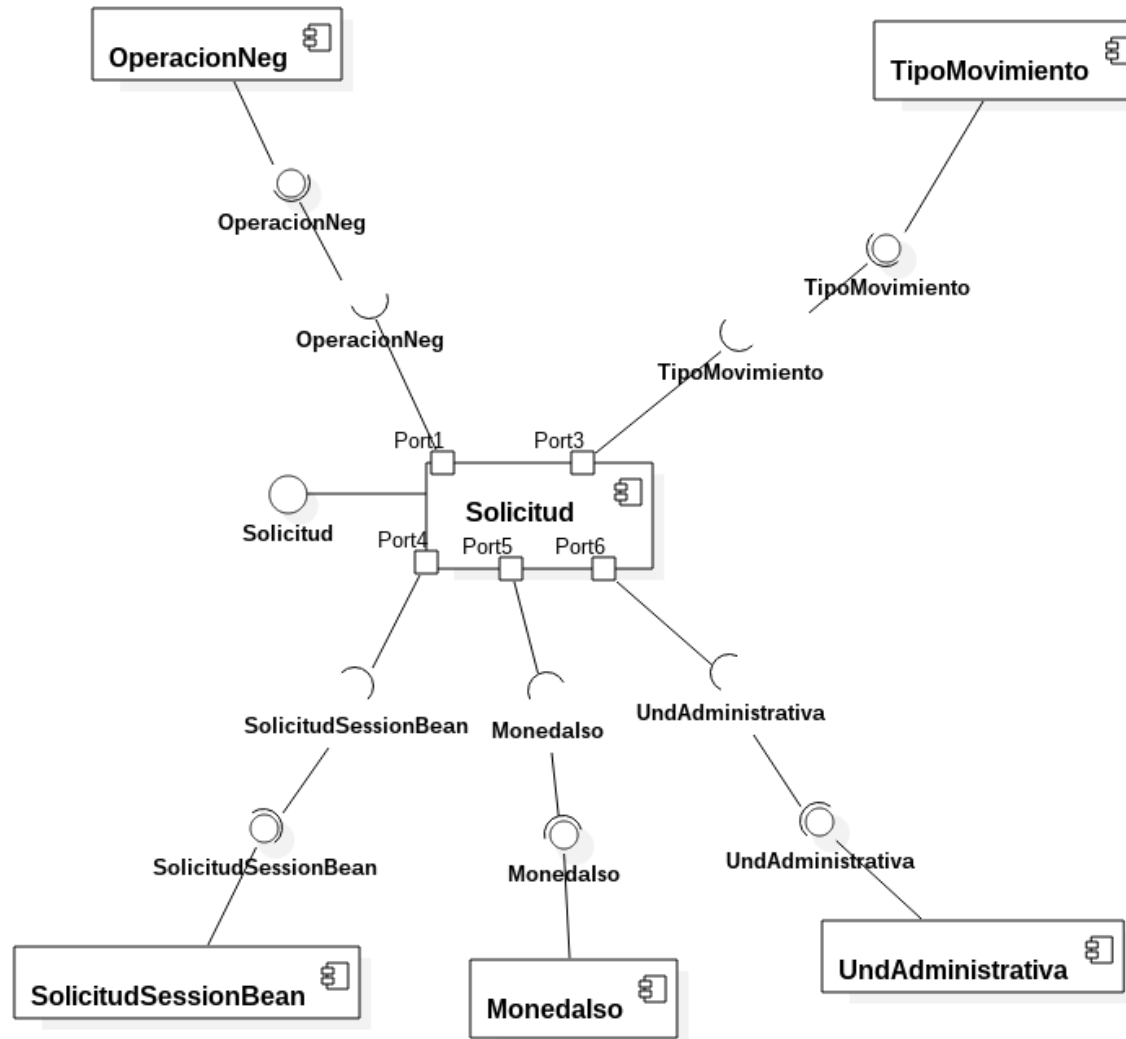


Ilustración 22 Componente de servicio. Catalogo



*Ilustración 23 Componente de servicio. Instrumento*



*Ilustración 24 Componente de servicio. Solicitud*

## Contrato de servicios

- **Descripción general de los servicios web**

Los servicios provistos están enfocados en la administración de los portafolios de instrumentos y en el proceso de conciliación de las operaciones que realiza la tesorería internacional. Cada servicio espera recibir y provee los datos en un formato *media type* Json o XML con los estándares que establece la industria. La localización de los recursos es a través del protocolo HTTPS se encuentran en <https://arti.bcv.org.ve/TesoreríaServer/webresources>

Como se ha mencionado con anterioridad se van a describir las operaciones de los servicios más importantes para las funcionalidades provista por los servicios para efecto de atenuar la fatiga que pudiese tener el lector y no exceder en el número de páginas para este trabajo especial de grado.

- **Descripción de las operaciones**

*Tabla 105 Eventos por servicios. Catálogo*

Recurso		Método	Descripción	Códigos de estados	Representación de entrada	Representación de salida
	/Catalogo	POST	Agrega una nueva jerarquía o ruta al catálogo de instrumentos.	201 400 500	CatalogosBeanXml	InstrumentoBeanXml
/Catalogo	PUT	Actualiza la jerarquía o ruta del catálogo. Puede ser actualizado por nodo o por toda la jerarquía del árbol.	202 400 500	CatalogosBeanXml	InstrumentoBeanXml	



Tabla 106 Eventos por servicios. Catálogo

<b>Recursos</b>	/Catalogo/catalogos	GET	Sustrae toda la jerarquía del catálogo correspondiente a los instrumentos.	200 500		CatalogosBeanXml
	/Catalogo/catalogos/catalogoComposition	GET	Es la operación encargada de orquestar la invocación de los servicios que proveen la lista de corresponsales por estatus, la lista de los tipos de instrumentos, la lista de los tipos de mercado de divisas y ejecuta la operación de la búsqueda de catálogos.	200 500		CatalogoCompositionBeanXml
	/Catalogo/catalogos/byNuCorrespCta/{nuCorrespCta}	GET	Se obtiene toda la rama del catálogo donde se ubique el instrumento asociado a la cuenta corriente del corresponsal.	200 500	String	CatalogosBeanXml

Tabla 107 Eventos por servicios. Instrumento

Recurso		Método	Descripción	Códigos de estados	Representación de entrada	Representación de salida
	/Instrumento/{n uInstrumento}	PUT	Actualiza los datos asociados al instrumento por el código del instrumento suministrado.	202 400 500	short	InstrumentoBeanXml
	/Instrumento	POST	Agrega un nuevo instrumento según los datos que se suministren.	201 400 500	InstrumentoBeanXml	InstrumentoBeanXml
	/Instrumento/movimiento	POST	Agrega un nuevo movimiento originado por el traspaso de fondos entre los instrumentos.	201 400 500	MovimientosBeanXml	MessageBeanXml

Tabla 108 Eventos por servicios. Instrumento

		Método	Descripción	Códigos de estados	Representación de entrada	Representación de salida
Recurso	/Instrumento/instrumentos/byNulInstrumento/{nInstrumento}	GET	Devuelve los datos del instrumento obtenidos del código del instrumento que se suministre.	200 500	short	InstrumentoCompositionBeanXml
	/Instrumento/instrumentos/byNuCorrespCta/{nuCorrespCta}	GET	Retorna todos los instrumentos que tengan asociado el código de la cuenta corriente del corresponsal.	200 500	String	InstrumentosBeanXml
Recursos	/Instrumentos/instrumentos/instrumentComposition	GET	Es la operación encargada de la orquestación de los servicios que suministra el listado de los tipos de instrumentos y busca toda la jerarquía del catálogo de instrumentos.	200 500		InstrumentoInitCompositionBeanXml

	/Instrumento/instrumentos/byRange	GET	Retorna el listado de instrumentos según los parámetros que se suministren. Todos los parámetros son opcionales.	200 500	String, String, short, String	InstrumentosBeanXml
	/Instrumento/instrumentos/excel	GET	Descarga del reporte de los montos no ejecutados que se encuentren entre los parámetros suministrados. Todos los parámetros son opcionales.	200 500	String, String, short, String	File
	Instrumento/instrumentos/movimientos	GET	Devuelve los movimientos asociados a los instrumentos según los parámetros de búsqueda que se suministren.	200 400	String, String, Short	MovimientosBeanXml

Tabla 109 Eventos por servicios. MsgSwift

Recurso	Método	Descripción	Códigos de estados	Representación de entrada	Representación de salida
MsgSwift/msgSwifts/startUpdater/{hour}	PUT	Establece la hora de ejecución del proceso de conciliación de las operaciones. La hora suministrada debe ser en formato de 24h.	201 400 500	String	MessageBeanXml

/MsgSwift/msgSwifts/byRange	GET	Suministra todos los mensajes SWIFT según los parámetros de búsqueda. Todos los parámetros son opcionales.	200 500	String, String, String, String	MsgSwiftBeanXml
/MsgSwift/msgSwifts/excel	GET	Descarga el reporte en Excel de todas las operaciones no identificadas según los parámetros de búsqueda. Todos los parámetros son opcionales.	200 500	String, String, String, String	ResponseBuilder
/MsgSwift/msgSwifts/byCodigoMsgSwift	GET	Retorna los datos del mensaje SWIFT por el código del mensaje suministrado.	200 500	String	MsgSwiftBeanXml
/MsgSwift/msgSwifts/processSwift	GET	Ejecuta el proceso de conciliación al invocar el servicio.	200 500		MessageBeanXml

Tabla 110 Eventos por servicios. Solicitud

Recurso		Método	Descripción	Códigos de estados	Representación de entrada	Representación de salida
	/Solicitud	POST	Permite añadir una nueva solicitud de cobertura.	201 400 500	SolicitudBeanXml	MessageBeanXml
	/Solicitud/solicitudes/startUpTime/{hour}	PUT	Establece la hora automática de ejecución en que el proceso de conciliación se efectuará.	202 500	String	MessageBeanXml
	/Solicitud/solicitudes/composition/new	GET	Es la operación encargada de orquestar el llamado a los servicios que suministran la lista de las monedas ISO disponibles y el listado de las unidades administrativas.	200 500		solicitudCompositionBeanXml
	/Solicitud/solicitudes/composition/init	GET	Es la operación encargada de orquestar el llamado a los servicios que proveen la jerarquía de los catálogos de los instrumentos y los tipos de movimientos.	200 500		SolicitudCompositionInitBeanXml

Recursos	/Solicitud/solicitudes/composition/mne	GET	Es la operación encargada de orquestar los servicios que proveen la jerarquía con los instrumentos que tienen asociado operaciones que no fueron ejecutadas y el listado de los tipos de movimientos.	200 500		solicitudCompositionInitBeanXml
	/Solicitud/solicitudes/s/isReconciliation	GET	Ejecuta inmediatamente el proceso de la conciliación de las operaciones que estén hasta el momento sin conciliar.	200 500	String, String	MessageBeanXml
Recursos	/Solicitud/solicitudes/montoNoEjecutado	GET	Suministra el listado de las solicitudes de coberturas que contengan operaciones que no fueron ejecutadas según los parámetros de búsquedas indicados. Todos los parámetros son opcionales.	200 500	String, String, short	SolicitudesBeanXml
	/Solicitud/solicitudes/excel	GET	Descarga el reporte en Excel de las operaciones que no fueron ejecutadas con base a los parámetros de búsquedas indicados. Todos los parámetros son opcionales.	200 500	String, String, short	ResponseBuilder

## Requisitos no funcionales de los servicios

Al diseñar una arquitectura orientada a servicios no solo se debe basar en las funcionalidades que deba implementar los servicios, se deben tener en cuenta la calidad de los servicios, es decir, los requerimientos no funcionales que deben tener.

Es importante especificar de manera clara cuáles son los requisitos no funcionales que satisfagan los servicios para cuando se diseñen y construyan garanticen estos atributos de calidad. En ese sentido se listan los atributos de calidad que los servicios soportaran:

- **Adecuación Funcional:** El producto de software debe proporcionar las funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas en el diseño.
- **Compatibilidad:** El sistema deberá poder intercambiar información y/o llevar a cabo sus funciones requeridas cuando deba comunicarse con otro sistema externo.
- **Usabilidad:** El producto de software debe ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se use bajo determinadas condiciones.
- **Fiabilidad:**
  - **Disponibilidad:** El sistema debe ser capaz de estar operativo y accesible para su uso cuando se requiera
- **Mantenibilidad:** El producto de software debe ser modificable efectiva y eficientemente que permita un cambio en un componente que tenga impacto mínimo sobre el resto y que tenga la capacidad de ser un activo reutilizable.

## Fase 4. Realización de servicio

A este punto justo queda hacer la toma de decisiones de realización de servicios, estando asociadas con componentes de servicio. Cada componente de servicio puede ser considerado un contenedor de funcionalidades por lo que es importante decidir cómo se ejecutaran estos componentes de servicio.

Las consideraciones que se tuvieron en cuenta para la realización de los servicios se ilustran en la siguiente tabla:



Tabla 111 Decisiones de realización de servicio

Componente de empresa	Servicio realizado	Componente funcional y técnico	Decisión de realización	¿Por qué?
Balances SWIFT	<p>Procesar SWIFT</p> <p>Consultar balances</p> <p>Conciliar operaciones</p> <p>Generar reporte</p>	<p>Descomponer los SWIFT en sus segmentos</p> <p>Consulta de balance</p> <p>Ubicar los montos no identificados.</p> <p>Conciliar las operaciones notificadas y ejecutadas</p>	<p>Fragmentar y almacenar los balances para su consulta e identificar de manera sencilla las operaciones</p>	<p>Al realizar el servicio permitirá la consulta de operaciones de forma general o detalladas e identificar las operaciones ejecutadas permitiendo ubicar los montos no identificados y conciliar las operaciones notificadas contra su parte ejecutada</p>
Solicitud de cobertura	<p>Crear cobertura</p> <p>Consultar coberturas</p> <p>Generar reporte</p> <p>Establecer la hora de ejecución de la conciliación</p>	<p>Generación de la cobertura</p> <p>Consultar las coberturas</p> <p>Identificar los montos no ejecutados</p> <p>Hora de ejecución de la conciliación</p>	<p>Recibir las solicitudes de coberturas de manera estructurada que permita consultar e identificar posteriormente las operaciones no ejecutadas</p>	<p>Tendiendo de manera organizada y bien estructurada las coberturas el servicio proveerá a las aplicaciones la consulta de las solicitudes de coberturas generadas.</p>
Catálogo de instrumentos	<p>Crear catálogo e instrumentos</p> <p>Modificar el catálogo y/o instrumento</p> <p>Consultar el catálogo</p>	<p>Registro de catálogo e instrumento</p> <p>Modificación</p> <p>Recuperación del catálogo</p>	<p>Organizar y estructurar el portafolio de instrumentos de manera jerárquica permitiendo crearlos y consultarlos</p>	<p>Estructurando los instrumentos se podrá saber de manera rápida la posición de los flujos de fondos de los corresponsales</p>

## Asignación de los servicios a los componentes

Deben tomarse decisiones sobre qué servicios se asignarán a qué componentes del servicio.

*Tabla 112 Asignación de los servicios a los componentes*

Servicio	Componente de servicio	Comentario
Catalogo	CatalogoController	El servicio provee las funcionalidades de crear y obtener el portafolio de instrumentos a través del componente controlador
Solicitud	SolicitudController	Administra todas las solicitudes de cobertura
MsgSwift	MsgSwiftController	Descompone e identifica las operaciones que no fueron notificadas

## Decisiones arquitectónicas relacionadas con la realización del servicio

Ciertos puntos de decisiones arquitectónicas pertenecen a los servicios y la capa de servicios. Estas decisiones se resumen en la siguiente tabla:

*Tabla 113 Decisiones arquitectónicas relacionadas con los servicios*

Punto de decisión	Decisión	Servicios
Cómo se externalizarán las descripciones de los servicios	WADL	Todos
Cómo se exponen los servicios	Web Services	Todos
Cómo se forman los mensajes	XML/JSON	Todos
Cómo se realizará la transformación de los mensajes	Java Jersey	Todos

*Tabla 114 Decisiones arquitectónicas relacionadas con los servicios*

Punto de decisión	Decisión	Servicios
Cómo se aplicará la seguridad en los servicios	No aplica	No aplica
Problemas de espacio de nombres de servicio	No aplica	No aplica
División de responsabilidad entre el ESB y los componentes empresariales	Componentes empresariales	Todos

# CAPITULO IV

## Implementación y despliegue

### Fase 5. Implementación

Conociendo el proceso de negocio y establecido los requerimientos, identificados y especificados los servicios para la construcción de la solución propuesta se empieza a desarrollar según el nivel de importancia cada componente y servicio que den soporte a los casos de uso planificados y a las decisiones de realización de los servicios.

Los beneficios de una arquitectura orientada a servicios como la flexibilidad de los componentes, la interoperabilidad y la reusabilidad están basados en un modelo en donde intervienen tres actores importantes: el proveedor de servicios, el consumidor de servicios y el registro del servicio Siempre previendo en que los procesos dentro de un negocio van a cambiar con el tiempo y gracias al uso de servicios estos cambios se manejan de una mejor manera y con un menor impacto, la construcción del sistema se llevó a cabo bajo los siguientes criterios:

- El desarrollo de los servicios se implementa con el conjunto de herramientas con lenguaje Java y Jersey con la implementación de referencia JAX-RS
- Con ello, se implementó los servicios que proveerán las funcionalidades del sistema; implementando primero los servicios de tarea, luego,
- La prueba de los servicios a nivel de construcción se realizará utilizando la herramienta SoapUI, posteriormente,
- Para el manejo y descomposición de los mensajes SWIFT se desarrollará utilizando el API java WIFE.
- El ambiente de trabajo para el consumo de data de prueba que sea la más exacta a la real tanto para los balances SWIFT y solicitudes de cobertura se usaran con data fechada de hace más de cinco años por temas de seguridad del banco.
- Se construirá la interfaz gráfica de usuario como aplicación web que consume los servicios y permita el manejo de los instrumentos y la consulta de reportes.
- El desarrollo de la aplicación web fue luego de tener consolidado los servicios de tarea.
- Consolidado todos los servicios se realizará pruebas de integración con los otros sistemas del banco para corroborar que efectivamente cumple con las necesidades funcionales.

## Estándares para la codificación en Java

En cualquier proyecto en el que intervenga más de una persona se hace necesario seguir unas guías comunes de desarrollo para asegurar la correcta comprensión de todo el código, evitar malas prácticas, y así como su mantenimiento posterior por personas que no han participado en el desarrollo, e incluso para quienes participaron pero retoman el proyecto mucho tiempo después es necesario tener estas convenciones de código.

Por ello, para la programación y/o codificación de los servicios, componentes y la aplicación web se realizaron en sintonía de los estándares de programación de la Coordinación Funcional de Operaciones Internacionales (CFOI), departamento adscrito a la Gerencia de Sistema e Informática.

### Paquetes de código fuente

Los paquetes de código Java para los servicios se realizará con la siguiente estructura:

- `ve.org.bcv.<componente>.<clase>`

Los paquetes de código Java para la aplicación web (jsp, servlets) se agruparan en carpetas según su principal características y/o funcionalidad. De tal manera que se organizaran como:

- Dashboard
  - Catalogo
  - Corresponsal
  - Solicitud
  - Reporte
  - Configuraciones
  - Transferencias

El paquete de servicios y el paquete de la aplicación web serán independientes, es decir, estarán las clases y componentes en paquetes diferentes con el fin de que se puedan desplegar de manera individual con el beneficio que esto trae.

Toda clase será comentada según la herramienta de JavaDocs que permite comentar de manera estructurada y/o organizada el código fuente.

### Clases e interfaces

- Los nombres deben ser en sustantivo y en singular
- Deben ser representados en CamelCase
- La clase debe llevar un sufijo identificativo según su funcionalidad

- Máximo 80 caracteres por línea
- Usar líneas en blanco como elemento de separación de bloques
- Debe haber solo una clase o interfaz por archivo
- Sangrar adecuadamente el bloque de código

Para el resto de elementos que conforman una clase se debe tener presente la guía de estilo de programación Java proporcionada por Google Inc.

## Herramienta para la implementación del proceso de negocio

Para la construcción y pruebas de los servicios se instaló y configuró el servidor de ambiente de desarrollo en WildFly Application Server y el JBossDeveloper. El servidor de aplicaciones quien se encargará de contener y proveer tanto los servicios que interactuarán con los otros sistemas del instituto como la aplicación web.

Ésta herramienta manejada por el Departamento de Programación y Control de Servicios (DPCS) adscrita a la Gerencia de Sistema e Informática son los encargados de administrar y configurar el ambiente de desarrollo, certificación y puesta en producción

La aplicación web que constituye la interfaz de usuario de las actividades que requieren la interacción con la administración de los corresponsales e instrumentos y sus reportes entre otros fue desarrollada con el JBossDeveloper.

## Implementación del proceso de negocio Conciliar Operaciones

La implementación de los servicios que soportaran el proceso de negocio Conciliar Operaciones se inició desarrollando los servicios de tarea, empezando por la construcción del servicio MsgSwift que se encarga mediante el API Java WIFE descomponer los mensajes Swift MT950 y MT940 que están estructurados como se muestra en la siguiente tabla:

*Tabla 115 Formato de descripción SWIFT MT950*

Etiqueta	Nombre de campo	Presencia	Comentario
:20:	Número de referencia de la transacción	Obligatorio	
:25:	Identificación de la cuenta	Obligatorio	Este campo especifica la cuenta por el cual fue enviado

:28c:	Número de declaración/Número de secuencia	Obligatorio	
:60:	Apertura del balance		60F = inicio de la apertura del balance 60M = Intermedio de la apertura del balance
	Símbolo de Crédito/Debito	Obligatorio	C = Crédito D = Débito
	Sentencia de fecha	Obligatorio	Formato = YYMMDD
	Moneda	Obligatorio	Código de moneda ISO
	Monto	Obligatorio	Apertura del balance con coma como un punto decimal
:61:	Línea de declaración	Opcional	Éste registro es producido por cada transacción en la cuenta declarada
	Fecha Valor	Obligatorio	Formato = YYMMDD
	Símbolo de crédito débito	Obligatorio	C = Crédito D = Débito
	Monto	Obligatorio	Monto de la transacción con coma como punto decimal
	Tipo de transacción Código de identificación	Obligatorio	N o F Para entradas de crédito el rango de 100-999
	Número transacción	Obligatorio	Número de referencia interna del banco
	Detalles complementarios	Obligatorio	Información adicional del banco en una nueva línea
:86:	Información del dueño de la cuenta	Opcional	Contiene detalles de cada transacción
:62:	Cierre del balance (fondos consignados)	Obligatorio	62F = Final del cierre del balance 62M = Intermedio del cierre del balance
	Símbolo de crédito/Debito	Obligatorio	C = Crédito D = Débito
	Declaración de fecha	Obligatorio	Formato YYMMDD
	Moneda	Obligatorio	Código de moneda ISO
	Monto	Obligatorio	Cierre del balance con coma

			como punto decimal
:64:	Cierre del balance disponible	Obligatorio	Fondos disponibles
	Símbolo de Crédito/Débito	Obligatorio	C = Crédito D = Débito
	Declaración de fecha	Obligatorio	Formato YYMMDD
	Moneda	Obligatorio	Código de moneda ISO
	Monto	Obligatorio	Cierre del balance disponible con coma como punto decimal

Conociendo cada extracto del balance se extrae y se procede a registrar en el gestor de base de datos.

Se realiza las consultas según la especificación de los servicios que mostrará el listado de los balances extraídos y generar la operación de generación del reporte en Excel con el uso del paquete Java Apache POI.

Para el resto de los servicios de tarea se ejecuta el mismo proceso, salvo que para el caso de los servicios compuesto que contiene el flujo y la secuencia de actividades e invocación de servicios que permitirán la orquestación de los servicios en tiempo de ejecución.

Entonces, con los modelos de los servicios se implementan cada uno de los servicios. Ya consolidados se procede a la construcción de la aplicación web que consumirá los servicios.

## **Desarrollo de la aplicación web asociados a la interfaz de usuario del proceso**

Una vez realizado la implementación de los servicios se procedió al desarrollo de la aplicación web, el cual será la interfaz de usuario requerida para que el analista financiero pueda llevar a cabo las tareas que requieran su interacción en el proceso de conciliación.

La aplicación web desarrollada cubre cada una de las funcionalidades del proceso de conciliación.

### **Descripción de la aplicación web**

Los módulos que a continuación se listan dan una breve descripción de su funcionalidad con respecto al proceso de conciliación.

Tabla 116 Módulos de la aplicación web

Módulo		Descripción del módulo
<b>Gestionar Corresponsales y cuentas corrientes</b>	Añadir, modificar corresponsales y cuentas corrientes	A través de unos formularios el módulo permite agregar toda la información relacionada a un corresponsal para almacenar los datos suministrados por el analista financiero.
	Listar corresponsales	Mediante un formulario con unos parámetros de búsqueda extrae los corresponsales y sus cuentas asociadas.
<b>Gestionar Instrumentos</b>	Lista el portafolio o catálogo, añade y modifica los instrumentos.	Permite al analista financiero crear, modificar instrumentos y visualizar de manera de una jerarquía de directorio la organización del catálogo del portafolio de instrumentos y ver sus datos y movimientos asociados
<b>Transferencias entre instrumentos</b>	Transfiere fondos entre los instrumentos	Permite realizar una transferencia de fondos a través de un formulario entre cuentas mediante el manejo de los instrumentos.
<b>Gestionar Solicitudes</b>	Añadir solicitud de cobertura	El analista financiero a través de un formulario puede crear una solicitud de cobertura propia de la tesorería y a su vez generarla para las áreas de negocio del instituto.
	Listar solicitudes	Permite buscar las solicitudes generadas por las áreas de negocio y gestionar el estatus de la misma.
<b>Gestionar reportes</b>	Generar reporte de montos no identificados	Con el uso de un formulario permite buscar por unos parámetros de búsqueda los montos no identificados y opcionalmente genera la descarga en Excel el reporte de los montos no identificados.
	Generar reporte de montos no ejecutados	Con el uso de un formulario permite buscar por unos parámetros de búsqueda los montos no ejecutados y opcionalmente genera la descarga en Excel el reporte de los montos no ejecutados.
	Generar reporte de montos reales y montos estimados	Con el uso de un formulario permite buscar por unos parámetros de búsqueda el reporte como resultado de la conciliación los montos reales y estimados y opcionalmente genera la



		descarga en el Excel el reporte
<b>Gestionar configuraciones</b>	Configurar elementos asociados a la aplicación	Agrega o modifica opciones de configuración de la aplicación web

### **Especificación de los módulos**

- **Descripción del módulo añadir corresponsales**

Con la introducción de los datos requeridos por el servicio S05P01 por el analista financiero mediante un formulario se invoca el servicio mediante AJAX y se envía la data en formato JSON.

### **Casos de uso relacionados**

- Caso de Uso UC04: Añadir Corresponsal
- Caso de Uso UC06: Gestionar Cuenta Corriente

### **Diagrama de secuencia relacionado**

Diagrama de Secuencia SD01: Insertar Corresponsal

- **Descripción del módulo añadir instrumentos**

Con la introducción de los datos requeridos por el servicio S03P01 por el analista financiero mediante un formulario se invoca el servicio mediante AJAX y se envía la data en formato JSON.

### **Casos de uso relacionados**

- Caso de Uso UC03: Agregar Instrumento

### **Diagrama de secuencia relacionado**

Diagrama de Secuencia SD01: Insertar Instrumento

- **Descripción del módulo transferencia**

Con la introducción de los datos requeridos por el servicio S03P05 por el analista financiero mediante un formulario se invoca el servicio mediante AJAX y se envía la data en formato JSON.

### **Casos de uso relacionados**

Caso de Uso UC03: Modificar Instrumento

### **Diagrama de secuencia relacionado**

Diagrama de Secuencia SD01: Modificar Instrumento

- **Descripción del módulo listar movimiento**

Con la introducción de los parámetros de búsqueda por el analista financiero mediante un formulario se invoca el servicio S09P02 mediante AJAX y se envía la data en formato JSON.

#### **Casos de uso relacionados**

- Caso de Uso UC03: Buscar Movimientos

#### **Diagrama de secuencia relacionado**

Diagrama de Secuencia SD01: Buscar movimiento

- **Descripción del módulo listar solicitudes**

Con la introducción de los parámetros de búsqueda por el analista financiero mediante un formulario se invoca el servicio S13P06 mediante AJAX y se envía la data en formato JSON.

#### **Casos de uso relacionados**

Caso de Uso UC03: Buscar Solicitud

#### **Diagrama de secuencia relacionado**

Diagrama de Secuencia SD01: Buscar Solicitud

- **Descripción del módulo listar montos no identificados**

Con la introducción de los parámetros de búsqueda mediante un formulario se invoca el servicio S13P06 mediante AJAX y se envía la data en formato JSON.

#### **Casos de uso relacionados**

- Caso de Uso UC03: Buscar Montos No Identificados

#### **Diagrama de secuencia relacionado**

Diagrama de Secuencia SD01: Buscar Montos no Identificados

- **Descripción del módulo gestionar configuraciones**

Permite agregar o eliminar los correos electrónicos, el asunto y contenido a usar para las notificaciones de las operaciones mediante un formulario. Se invoca el servicio S13P06 mediante AJAX y se envía la data en formato JSON.

#### **Casos de uso relacionados**

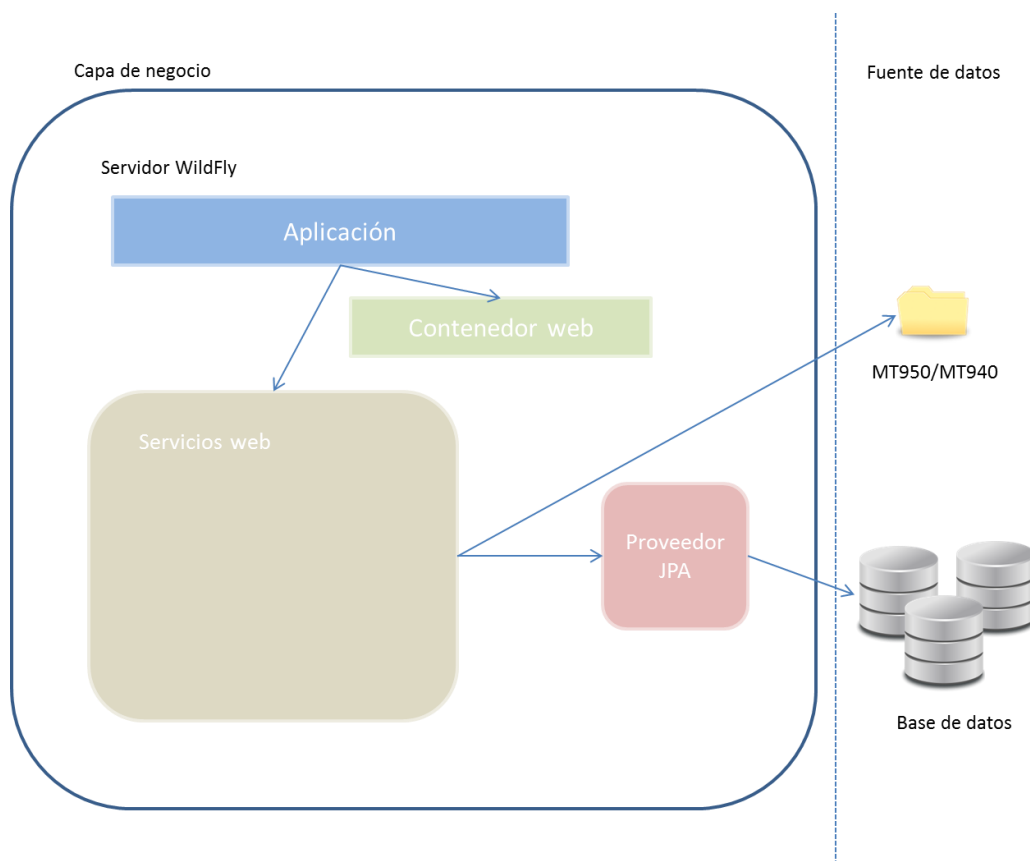
- Caso de Uso UC03: Gestionar Correo
- Caso de Uso UC04: Gestionar Documento HTML

#### **Diagrama de secuencia relacionado**

Diagrama de Secuencia SD01: Insertar Correo

## Fase 6. Despliegue y pruebas

La fase previa de implementación contiene los elementos para la instalación y configuración de los componentes e interfaces de los servicios para ser desplegados en entorno de producción. El despliegue de la SOA es implementado sobre la plataforma JEE de WildFly donde la aplicación web va a estar en el *web container* y los servicios web bajo una capa de negocio soportada por el contenedor de servicios, como se muestra en la Figura.



*Ilustración 25 Vista del despliegue SOA en la plataforma*

Esta fase de despliegue, integración y pruebas estuvieron enfocadas en la publicación de los componentes desarrollados a través de sus interfaces, para ejecutar sobre ellos las pruebas pertinentes. En esta fase el objetivo fue el de la realización de las pruebas dirigidas a determinar si el comportamiento de la interfaz y la información compartida entre los servicios cumplía con los requerimientos especificados. El objetivo fue el de garantizar que todos los servicios cumplen con lo especificado en la definición de la interfaz, en término de normas, la validación del formato y los datos.

La primera meta conduce a la prueba de validación; en ella, se espera que el sistema se desempeñe de manera correcta mediante un conjunto dado de casos de prueba, que refleje el uso previsto del sistema.

La finalidad de la verificación es comprobar que el software cumpla con su funcionalidad y con los requerimientos no funcionales establecidos. Sin embargo, la validación es un proceso más general. La meta de la validación es garantizar que el software cumpla con las expectativas del cliente.

Al probar el software a nivel de clases y/o objetos, se ejecuta un programa con datos artificiales con la utilización de la herramienta JUnit, esto enmarcado en el plan de pruebas unitarias.

### **Plan de pruebas unitarias**

Las pruebas de unidad son el proceso de probar el programa, como métodos o clases de objetos. Las funciones o los métodos individuales son el tipo más simple de componente. Las pruebas deben llamarse para dichas rutinas con diferentes parámetros de entrada. Los datos de prueba son las entradas que se diseñaron para probar un sistema. Los casos de prueba son especificaciones de las entradas a la prueba y la salida esperada del sistema (los resultados de la prueba).

Para ello se utilizó la librería JUnit de Java. JUnit, una librería desarrollada para poder probar el funcionamiento de las clases y métodos que componen nuestra aplicación que permite asegurarnos de que se comporta como debe ante distintas situaciones de entrada, estas situaciones de entrada serían los casos de pruebas.

Para el plan de pruebas unitarias se diseñaron con los siguientes criterios:

- Se prueba todas las operaciones asociadas con el objeto,
- Estableciendo y verificando el valor de todos los atributos relacionados con el objeto, y
- Poniendo el objeto en todos los estados posibles

A través de varias iteraciones se fueron haciendo las correcciones en las clases según los resultados arrojados por las pruebas unitarias hasta llegar al 100% de casos de pruebas exitosos.

### **Plan de pruebas de componentes**

Las pruebas de componentes es donde muchas unidades individuales se integran para crear componentes compuestos. Las pruebas de componentes se enfocan en probar interfaces de los componentes para demostrar que se comporta según su especificación.

En este sentido este plan de pruebas de componentes se diseñaron siguiendo algunos lineamientos generales para las pruebas de las interfaces tales como:

- Un conjunto de pruebas donde los valores de los parámetros hacia los componentes estuviesen en el rango de sus valores extremos,
- Donde los valores fuesen nulos,
- Donde deliberadamente falle el componente, es decir, esperando generar una excepción controlada,
- Finalmente generando casos de pruebas extremos donde los pases de mensajes requieran un gran esfuerzo mucho mayor que el que realmente pueda ocurrir en la práctica.

Según el caso se fueron corrigiendo y mejorando los componentes para que cumpliera con el 100% de casos exitosos que fueron elaborados.

### **Plan de pruebas de sistema**

Las pruebas de sistema es básicamente todos los componentes que constituyen el sistema que se integran como un todo y se prueba. Estas pruebas están enfocadas a las interacciones de los componentes y buscan demostrar que los componentes son compatibles y que interactúan de manera correcta transfiriendo los datos en el momento adecuado entre sus interfaces.

Para el diseño de estas pruebas se apoya del uso de los diagramas de secuencia para diseñar los casos de pruebas específicos necesarios que se ejecutaran ya que muestra cuales entradas se requieren y cuales salidas se esperan.

### **Plan de pruebas de usuario**

Finalmente se realizó las pruebas de usuarios, estas pruebas están enfocadas en la aplicación web donde los usuarios interactúan a través de la interfaz gráfica de usuario con el sistema. En general estas pruebas están enfocadas en comprobar la experiencia del usuario con el uso del producto de software para ver si les gusta, si les satisface, si es fácil de ser comprendido, usado, y si en definitiva, hace lo que requieren.

Para ello se hizo un plan de pruebas de aceptación, en este plan se estableció el tiempo o calendario para la ejecución de las pruebas tomando como referencia los requerimientos y estableciendo el orden en que se prueban las características del sistema.

El resultado de estas pruebas se discuten con el usuario con la finalidad de determinar si el sistema es suficientemente apto para desplegarlo en producción.

### **Cómo se va a probar**

Como se mencionó para las pruebas se ha usado JUnit y para la capa de los servicios web los casos de prueba se ha usado SoapUI. A continuación se define

la manera en que se usó las herramientas con el que se va a desarrollar las pruebas:

## JUnit

En cada prueba se realizará una operación y se comprobará que el resultado devuelto es el esperado, para ello, JUnit dispone de los siguientes métodos para hacer comprobaciones:

- **assertEquals(resultadoEsperado, resultado):** Verifica que el valor esperado sea igual al resultado del test. Lanza una excepción si no se produce el resultado esperado.
- **assertTrue(condición):** Verifica que la condición a evaluar sea verdadera. Lanza una excepción si no cumple la condición.
- **assertNotNull(objeto):** Verifica que objeto a evaluar no sea *null*

Todos los métodos anteriores admiten opcionalmente un argumento *String* con un mensaje explicativo de la comprobación que realiza el *test*. De estar presente, este argumento sería el primero a ser suministrado.

JUnit dispone de mucho otros métodos pero para efectos de las pruebas realizadas no fueron necesarios utilizarlos.

Para inicializar cada una de los casos de pruebas JUnit cuenta con las anotaciones que ejecutaran solo una vez al inicio de los *test* el método donde se vincule la anotación. Estas anotaciones son **@Before** y **@After**, a través de esas anotaciones en los métodos se inicializan las variables que se requieran para la ejecución del caso de prueba.

A continuación, se detallan las anotaciones que incluye la librería:

**@RunWith:** Se le asigna a la clase al que se va hacer el *test* con el valor *Parameterized.class*.

**@Parameters:** Anotación para un método que proporciona parámetros para ser inyectados en el constructor de la clase de prueba por parámetro.

**@Before:** Indica que el método se debe ejecutar antes de cada *test*.

**@After:** Indica que el método se debe ejecutar después de cada *test*.

**@Test:** Indica a JUnit que se trata de un método de *Test*.

Una vez creados todos los *test* se ejecutaron. En la tabla 270 se pueden observar algunos resultados obtenidos.

Tabla 117 Casos de pruebas con JUnit

Clase	Método	Resultado
CatalogoControllerTest	testInsert	Al introducir el identificador del elemento padre del instrumento no validaba que existiese el identificador suministrado originando un error. Se introdujo esa condición y retorna una excepción en caso de que no exista el identificador.
CorresponsalControllerTest	testGetByInVisual	Al hacer la búsqueda de un corresponsal recién creado no se conseguía entre el listado devuelto. Como la persistencia a la base de datos se delegó a través de JPA este mantenía en cache los datos en el PersistenceContext por lo que si se hacia otra instancia de la persistencia no encontraba el registro que se buscaba.
SolicitudControllerTest	testInsertDetailRequest	Al ingresar los detalles de la solicitud de la cobertura establecía erróneamente el monto declarado debido a una conversión del tipo de dato. Finalmente se estableció el tipo de dato a BigDecimal
InstrumentoControllerTest	testGetByNuCorrespCta	Al hacer la búsqueda de todos los instrumentos asociados a una cuenta corriente del corresponsal no retornaba los movimientos de los fondos originado por las operaciones.

## SoapUI

SoapUI es una herramienta desarrollada en java para la realización de pruebas a aplicaciones con arquitectura orientada a servicio permitiendo la creación y ejecución automatizada de pruebas de software tales como pruebas funcionales, pruebas de regresión, simulación de servicios, pruebas de estrés y de seguridad.

Las pruebas funcionales están orientadas a validar y mejorar la calidad de los *webservices* creando y ejecutando escenarios de pruebas complejos e inclusive ejecutándolos asíncronamente.

Para ello se generó el WADL que describe los servicios, y a través del SoapUI se importa el archivo y se generan los casos de pruebas.

Esta herramienta maneja el concepto de *TestSuite*, *TestCase*, *TestStep*. Un *TestSuite* sirve para contener un número arbitrario de casos de prueba (*TestCases*) que pueden ejecutarse secuencialmente o en paralelo. Los *TestSteps* sirven para ejecutar *TestCases* secuencialmente. Una vez creadas las peticiones, es posible generar una *TestSuite* y así automatizar las pruebas funcionales.

Por cada *TestCase* es posible hacer validaciones automáticas de los resultados. Entonces, por cada petición, se verifica si la respuesta es un fallo, o no lo es, o contiene un determinado valor, o no lo contiene.

Se puede observar en la tabla 118 algunos resultados obtenidos de los casos de prueba.

*Tabla 118 Casos de prueba con SoapUI*

Servicio	Recurso	Método	Resultado
Catalogo	/Catalogo	POST	Salía error no controlado si el atributo nuCatalogo y nuPredecesora eran recibido como String. El error era que no hacía la comprobación del tipo de dato recibo. Se solucionó añadiendo un ShemaValidator que verificara el tipo de dato suministrado.
Corresponsal	/Corresponsal /corresponsales/byCoCorresponsal/ {coCorresponsal}	GET	Si el nombre del corresponsal contenía caracteres especiales no retornaba bien el nombre. Se solucionó estableciendo todo el texto en UTF8
Solicitud	/Solicitud/solicitudes/excel	GET	Se generaba siempre el mismo reporte en Excel. Se solucionó estableciendo en el Header de la respuesta el cache control en null.
MsgSwift	/MsgSwift/msgSwifts/startUpTimer/{ hour}	PUT	Generaba un error no controlado si el formato de la hora no era correcto. Se solucionó añadiendo una expresión regular de @Pattern que comprobara el formato.

### **Aplicación Web implementada**

Para las pruebas de usuario o de aceptación consisten en que el usuario valide o compruebe y que finalmente acepte que el producto o aplicación se ajusta a los requisitos especificados por él a lo largo de todo el desarrollo del software. Este enfoque de calidad de la aplicación cubren los elementos tales como:



- **Contenido:** Se valora la gramática de la información presentada, la consistencia de la información y la relevancia de la misma.
- **Funcionalidad:** Se verifica que cada módulo de la aplicación tenga falta de errores y falta de conformidad con los requerimientos del usuario.
- **Estructura de la aplicación:** Se garantiza que el contenido y funcionalidades de la aplicación sea extensible o que pueda soportar futuros contenidos o funcionalidades.
- **Usabilidad:** Se asegura que el usuario tenga la capacidad de aprender de manera simple el uso de la interfaz gráfica de usuario y que soporte el rol de cada usuario.
- **Interoperabilidad:** Se garantiza que la aplicación web tiene la interfaz requerida para la conexión con los servicios web del sistema y que se comunica correctamente a través de AJAX.
- **Seguridad:** Se verifica que el usuario autenticado con sus credenciales sea el único que pueda tener acceso a la aplicación.

Para ello se elaboró un cronograma de uso de la aplicación por parte del usuario donde incluye la presentación y capacitación del sistema al usuario y posteriormente se elaboró una encuesta con una serie de preguntas referente a la calidad de la aplicación con el enfoque antes mencionado. Entre las preguntas de la encuesta están:

1. ¿El título de la ventana es claro y conciso?
2. ¿Se diferencian y están claro los campos obligatorios?
3. ¿La navegación en la pantalla cumple con la secuencia correcta?
4. ¿Los nombres de los campos indican claramente el fin que tiene?
5. ¿Los enlaces funcionan correctamente?
6. ¿Existe paginación adecuada para el número de registro que se consigan al hacer una búsqueda?
7. ¿Es relevante la información presentada?
8. ¿Funciona correctamente en los principales navegadores. Firefox y Chrome?
9. Funcionalidad de los módulos:
  - a. ¿Realiza la inserción de los datos?
  - b. ¿Realiza la modificación de los datos?
  - c. ¿Realiza la eliminación de los datos?
  - d. ¿Realiza la búsqueda de los datos y devuelve los datos esperados?
  - e. ¿Se ejecuta el reporte según corresponda?

Luego de unas iteraciones con el usuario se logró que el sistema cumpliera con todas las necesidades que requerían. Quedando finalmente la aplicación web como se ilustra a continuación:



Ilustración 26 Inicio de sesión

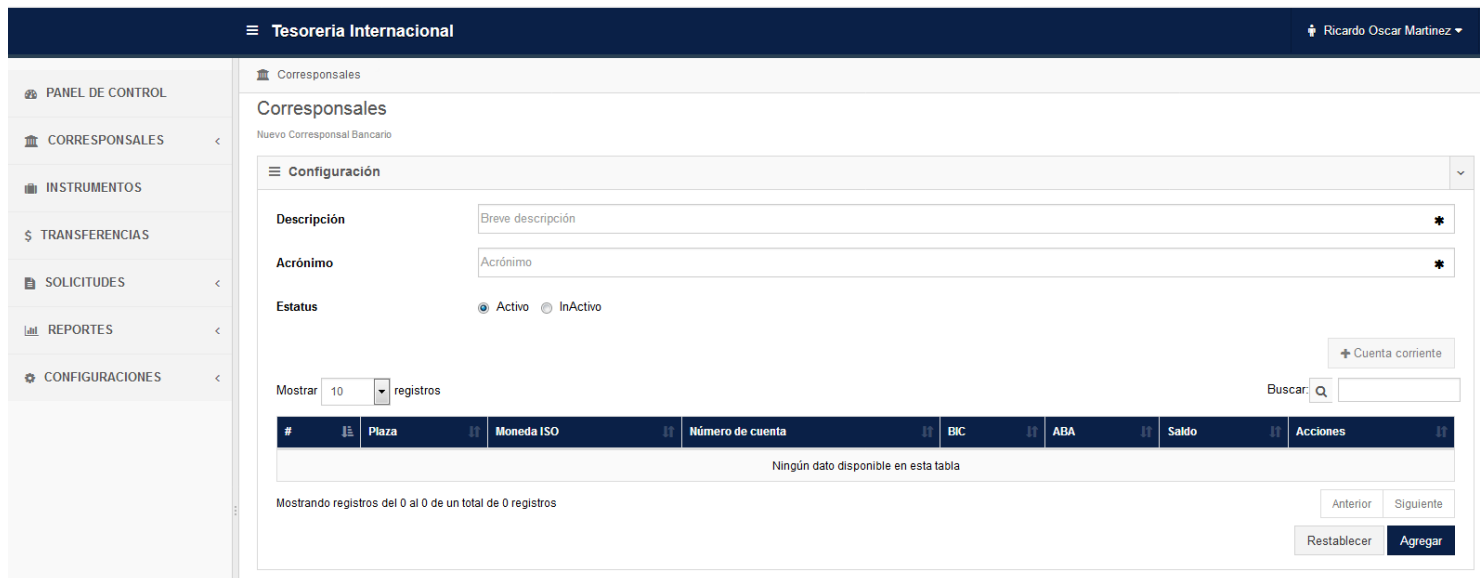


Ilustración 27 Creación de un corresponsal

**Tesorería Internacional** Ricardo Oscar Martínez

**Instrumentos**  
Catálogo de instrumentos

**Configuración**

**Instrumento**

- ROOT
  - BCV
    - RI TOTALES
      - RIO
        - CARTERAS DE INVERSI?N
          - BCV A
            - DEUTSCHE BANK A.G./MUNICH/EUR/1361000/EUR DEUTSBKCTPP/OFICIAL/1202017844601
            - DEUTSCHE BANK A.G./MUNICH/EUR/1361000/EUR DEUTSBKCTPP/DIPRO/1202017844601
          - BCV B
            - DEUTSCHE BANK A.G./FRANKFURT/EUR/1360100/EUR DEUTSBKDIV/OFICIAL/1202017844602
          - BCV C
            - DEUTSCHE BANK A.G./FRANKFURT/EUR/1360100/EUR DEUTSBKDIV/DIPRO/1202017844602
            - NUEVO CORRESPONSAL/BEIJING/CNY/1234556/NI/DICOM CC39/1242018059201
            - NUEVO CORRESPONSAL/BEIJING/CNY/1234556/NI2/DICOM CC38/1242018059201
          - BCV C DICOM-SIMADI
            - BCV C SICAD
            - BCV C SICADII
          - TR LARGO PLAZO
            - TRAMO A SICAD
            - TRAMO A SICAD II
          - TR DICOM SIMADI
            - OT DIV
            - OT DIV DICOM
              - DEUTSCHE BANK A.G./RUSIA/RUB/53484180/US\$ DEUTSBKDIV/DICOM CC38/1242017844602
        - TRAMO ESPECIAL
        - FONDO DEPARTAMENTOS
        - FONDO ESTABILIZACIÓN RESERVA
      - RNO
        - OTROS RNO
      - OAME

*Ilustración 28 Catálogo de instrumentos*

**Tesorería Internacional** Ricardo Oscar Martínez

**Solicitudes**

Nueva solicitud

**Configuración**

**Ingreso o Egreso de Divisas**

Ingreso     Egreso    \*

**Departamento Tramitador:** SELECCIONE \*      **Fecha elaboración:** 26-06-2018

**Departamento Solicitante:** SELECCIONE \*      **Persona contacto:** Persona contacto \*

**Área administrativa:** SELECCIONE \*      **Fecha Valor:** 26-06-2018

**Divisa:** SELECCIONE \*      **Banco corresponsal:** SELECCIONE \*      **Plaza:** SELECCIONE \*      **Cuenta corriente:** SELECCIONE \*

+ Detalle

Mostrar 10 registros      Buscar:

Tramo	Operación del negocio	Monto	Acciones
Ningún dato disponible en esta tabla			

Mostrando registros del 0 al 0 de un total de 0 registros      Anterior    Siguiente

**Observaciones:**

Observaciones \*

**Guardar**

*Ilustración 29 Solicitud de cobertura*

**Tesorería Internacional** Ricardo Oscar Martínez

**Reporte**  
Saldo: Montos no identificados

**Configuración**

**Busqueda**

Fecha valor de inicio:  Fecha valor final:

Mostrar  registros Buscar:

#	Cuenta corriente	Fecha Valor	Referencia	Monto	Credito/Debito	Acciones
FIN.9501803131249117	CITIBANK/NUEVA YORK/USD/36298291	27-03-2014	50RS201403240008 / BANCO PROVINCIAL S.A.BCO UNIVERSAL	5.390,78	CREDITO	
FIN.9501803131662482	CITIBANK/NUEVA YORK/USD/36298291	27-03-2014	50RS201403240008 / 2014032100037666ABC DO BRASIL LTDA	3.519,76	CREDITO	
FIN.9501803132502727	CITIBANK/NUEVA YORK/USD/36298291	27-03-2014	50RS201403240008 / ORG 312631 AMERICAN AIRLINES INC.	342.046,05	CREDITO	
FIN.9501803133070943	CITIBANK/NUEVA YORK/USD/36298291	27-03-2014	50RS201403240008 / ORG CITIVECA CITIBANK N.A SUCURSA	28.968,47	CREDITO	
FIN.9501803135708283	CITIBANK/NUEVA YORK/USD/36298291	27-03-2014	50RS201403240008 / BANCO PROVINCIAL S.A.BCO UNIVERSAL	11.000,00	CREDITO	

*Ilustración 30 Reporte: Montos no identificados*

# CAPITULO V

## Conclusiones

Al ser la metodología SOMA orientada a servicios se logró sistematizar cada tarea y actividad. Ello, permitió lograr los objetivos trazados en este Trabajo de Maestría, alcanzando describir todo el proceso realizado para el desarrollo del sistema. Gracias a la referida metodología se alcanzó una adecuada documentación de cada fase del desarrollo, haciendo registro de todas las actividades realizadas para lograr los objetivos. SOA-BPM facilitó la implementación de este Trabajo de Grado, debido a que los servicios y los procesos de negocio están estrechamente relacionados.

Culminado el desarrollo e implementación de la solución SOA, para el proceso de negocio de la conciliación de la tesorería internacional, descrito en este trabajo, le permitirá en el futuro a otros actores, desarrollar la ampliación de funcionalidades que requiera el sistema sin afectar los componentes que lo conforman. La implementación satisfactoria de la solución SOA para el proceso de conciliación hizo cimientos para una agregación de funcionalidades en los demás procesos del DITI. Desde un punto de vista funcional la solución implementada permitió al DITI mejorar el proceso de negocio de la conciliación, adecuándose a los requerimientos actuales del negocio y permitiendo la especificación detallada del flujo del proceso.

Los funcionarios del DITI, ahora pueden contar con un conjunto de funcionalidades de negocio, que ahora son expuestas como servicios y podrán ser consumidas y aprovechadas por los demás procesos de negocios del DITI y demás áreas que interactúan con ellos, facilitando el trabajo cotidiano que realizan.

La aplicación web realizada a través de Java EE representa un cambio de gran importancia para los usuarios, acostumbrados a una interfaz poco usable, que existe actualmente con el SATI. La adaptación del proceso de negocio y la incorporación de nuevas funcionalidades, actualmente es de menor impacto ya que el proceso cuenta con un menor acoplamiento entre sus componentes, ventajas que ofrece la arquitectura orientada a servicios.

La solución SOA implementada sobre WildFly y la aplicación Web para el proceso de la conciliación de la tesorería internacional le permite al DITI contar con un proceso adaptable y flexible a las necesidades de cambio del negocio, además de contar con un nuevo paradigma de arquitectura para la implementación de los demás procesos del DITI.

Una solución que involucra una arquitectura SOA en conjunto con BPM permite alinear los activos tecnológicos de la Coordinación Funcional de Operaciones

Internacionales con los procesos de negocios, conllevando a unos resultados y beneficios mayores a los actuales con un sistema monolítico como lo es el SATI. Los beneficios de la solución optimizan los procesos de negocios y reduce el acoplamiento entre los componentes por lo que las áreas de negocio pueden enfrentar los cambios sin tanto trauma y los desafíos que el departamento desempeña.

Con la categorización de los servicios se realizó la composición u orquestación de los servicios de manera que cada *web services* tuviese el engranaje necesario para suministra los datos que cada paso del proceso de negocio requiera.

A nivel técnico el estilo arquitectónico REST empleado, facilitó la interoperabilidad con los sistemas basados en PowerBuilder, éstos se integran mejor con el sistema invocando los servicios y recibiendo los datos en formato XML o Json, ésta interoperabilidad para el momento de este trabajo no está hecha en ambiente de producción. Junto al plan de pruebas desarrollados, se utilizó la herramienta SoapUI que permitió hacer ensayos exhaustivos a los servicios, que finalmente fueron expuestos. Lo anterior, facilitó la integración con los sistemas en PowerBuilder y se verificó el correcto manejo de las excepciones, dando un flujo alternativo al ocurrir un error al normal del proceso, ofreciendo una adecuada respuesta al usuario.

Aunque el trabajo realizado se enfocó en la identificación, especificación y realización de los servicios, el desarrollo de la aplicación web también fue parte importante en vista que mediante éste, los usuarios finales consumen los servicios y validan que las funcionalidades del sistema estén acorde a sus necesidades.

Finalmente, es importante señalar que el método SOMA tiene algunas otras actividades que no se realizaron o se adaptaron para el presente Trabajo de Maestría, pues el mismo está basada en RUP, que lo convierte en un método bastante pesado y se buscó la mayor flexibilidad posible para no generar documentación excesiva y poco relevante; tanto para el área de negocio como para el desarrollo del sistema.

## Referencias bibliográficas

Arellano J. (2015). Mejores Prácticas de Tesorería. Ponencia presentada en el 3er Seminario de Tesorería, México.

Arsanjani A. (2004). Service-Oriented Modeling and Architecture. How to identify, specify, and realize services for your SOA, IBM Journal.

Banco Central de Venezuela (2012). Manual de Organización y Funciones. Departamento de Inversiones y Tesorería Internacional

Grifols (2016). Caso de Estudio: Grifols: la conectividad y seguridad de SWIFT, claves para seguir creciendo. España.

López Martínez F. y Soriano Mas N. (2014). La gestión de la Tesorería. En qué Consiste y Cómo debe abordarse. España.

Sivakumar G. (2010). Fijación de los enfoques de la identificación de SOA/SOMA Service a los entornos del proyecto. IBM Journal.

Cervantes M. y Vergara J. (2012). Portafolios de inversión: Una alternativa para el Aprovechamiento de los Recursos Remanentes de Tesorería. Tesis de pregrado, Universidad del Rosario, Bogotá, Colombia.

Stephen A. y Miers D. (2009). BPMN. Guía de Referencia y Modelado. USA

Sommerville Ian (2011). Ingeniería de Software. Mexico

Dunnavant, T y Johnston, G. (2011) Design and develop a more effective SOA, Part 1: Introducing IBM's integrated capabilities for designing and building a better SOA, IBM Journal.

Earl T. (2009) SOA Manifiesto. [página web en línea] disponible: [http://www.soa-manifesto.org/default\\_spanish.html](http://www.soa-manifesto.org/default_spanish.html) [consulta: 2018, Mayo 20]

Earl T. (2008) SOA: Principles of Service Design. USA

Díaz Z (2006). Método para el Desarrollo de Servicios Web. Tesis de postgrado, Universidad de Los Andes, Mérida, Venezuela.

Navarro R. (2006). REST vs. Web Services. Universidad Técnica de Valencia

Gulabani S. (2013) Developing RESTful Web Services with Jersey 2.0. USA

Evans M. y Schoonderwoerd R. (2011) Modelación Orientada al Proceso para SOA, Parte 4: Relacionando todo con un estudio de caso. Cómo su modelo de proceso impulsa los modelos de casos de uso y servicios. IBM Journal

IBM Corp (2006) Modelado orientado a servicios y arquitectura en IBM [página web en línea] disponible: <https://cgrw01.cgr.go.cr/rup/RUP.es/index.htm> [consulta: 2018, Mayo 20]