Implementación de Servicios de integración web para Seguros Bolívar desarrollados por IBM

Santiago Andrés Cáceres Silva

Universidad Santo Tomas.
Facultad de ingeniería electrónica.
Proyecto de grado
Junio 2020.

Implementación de Servicios de integración web para Seguros Bolívar desarrollados por IBM

Trabajo de grado para obtener el título de:

Ingeniero Electrónico

Santiago Andrés Cáceres Silva

Director:
Ing. Daniel Alejandro Rodríguez Caro

Universidad Santo Tomas.
Facultad de ingeniería electrónica.
Anteproyecto de grado
Junio 2020

Exoneración De Responsabilidades

En el presente documento el autor se hace responsable de toda la información y afirmaciones que en él se realizan. La Universidad Santo Tomas no se hace responsable de las consecuencias que pueda generar este trabajo.

DEDICATORIA

El deseo final de presentar este trabajo de grado es obtener el título de ingeniero electrónico, el cual quiero dedicarlo especialmente a mi mamá Lyda Silva por su apoyo y dedicación a lo largo de estos años de estudio, a pesar de las dificultades y por siempre querer que con educación fuera una mejor persona, a Dios por darme Vida, salud y sabiduría para terminare esta etapa de mi vida.

AGRADECIMIENTOS

A dios por darme la sabiduría, y fortaleza para seguir en este camino.

A mi madre por el gran esfuerzo, apoyo, dedicación, motivación y por siempre creer en mi a lo largo de mi vida, por siempre enseñarme a que lo quiera lo logro.

A toda la facultad de ingeniería electrónica de la universidad Santo Tomas Tunja, por su gran labor a nivel regional.

A los docentes que me acompañaron en este proceso muchas gracias por compartir su conocimiento en las aulas de clase y por siempre buscar profesionales íntegros.

A mi tutor Ingeniero Daniel Caro, por ser la persona que me apoyo en este proceso de práctica, siempre orientándome y compartiendo información sobre alguna duda presentada.

A mis compañeros con los que se compartió momentos grandes momentos, de esfuerzo, de dedicación y en especial de amistad.

A IBM por la gran oportunidad de acompañarlos este tiempo aprendiendo cada día algo más de tecnología

NOTA DE ACEPTACIÓN[vi]

_____

_____

_____

_____

_____

_____

Firma Jurado

_____

Firma Jurado

_____

Firma Jurado

Tabla de Contenidos

Capítulo 1 Resumen 1
Capítulo 2 Introduccion 4
Capítulo 3 Problema. 4
Capítulo 4 Justificacion de la creacion de servicios web de integracion 5
Capítulo 5 Objetivos.. 6
5.1. General.. 6
5.2. Específicos.. 6
Capítulo 6 Marco Referencial. 7
6.1. Marco Teorico 7
6.2. ESB. 7
6.3. Arquitectura SOA. 8
6.4. Metodologia Scrum. 10
6.5. PL/SQL. 12
6.6. Stored Procedure 14
6.7. Git 14
6.8. Jira 15
6.9. Wsdl. 15
6.10. Xml. 15
6.11. Servicios Web. 15
6.12. Xsd. 15
Capítulo 7 Diseño Metodologico. 16
7.1. Metodología usada en el proyecto 16
7.2. Participantes. 17
7.3. Recursos Disponibles. 17
Capítulo 8 Desarrollo. 18
8.1. Requisitos 18
8.2 Asignacion del requerimiento en jira. 18
8.3. Creacion de servicio de integracion. 19
8.4. Creacion de adaptador 23
8.5. Elaboracion del contrato 31
8.6. Transformacion o mapeos. 33
8.7. Creacion del proxy service. 38
8.8.Generacion de jar y despliegue 42
8.9. Pruebas en SoapUI5.5.0. 46
8.10. Documentacion. 51
8.11. Solicitud de peer review. 52
8.12.Entrega 52
Capitulo 9 Conclusiones 55
Lista de referencias 57

Lista de figuras

Figura1. Patrón de arquitectura ESB……..….6

Figura2. Integracion de sistemas utilizando las capacidades de un ESB…….…...8

Figura3. Se presenta la comparacion entre SOA y sistemas monolíticos……..…11

Figura4. Representeacion de la metodología scrum…….……13

Figura5. Arquitectura PL/SQL……..……12

Figura6. Bloques PL/SQL…….…...…12

Figura7. Imagen de la metodología desarrollada…….……..…16

Figura8. Asignación del requerimiento en Jira…….……..…17

Figura9. Explicación del requerimiento en jira…….…..…18

Figura10. Grafico del requerimiento…….…..…18

Figura11. Icono del OSB……..….18

Figura12. Creación de configuración del OSB……..…19

Figura13. Configuración del OSB…….…19

Figura14. Configuración del OSB…….…20

Figura15. Creación de proyecto de OSB……...20

Figura16. Asignación de nombre al proyecto……....20

Figura17. Confirmación de creación del proyecto…….21

Figura18. Carpetas Creadas…….…..21

Figura19. Icono Jdevloper…….21

Figura20. Elección de SOA APLICATION……..22

Figura21. Elección de ruta del adaptador……..22

Figura22. Elección de parámetros predeterminados……..23

Figura23. Elección de database adapter……...23

Figura24. Conexión a base de datos……..…24

Figura25. Ingreso de parámetros de base de datos……...24

Figura26. Elección de stored procedure……...25

Figura27. Confirmación del stored procedure elegido…….25

Figura28. Visualización del adaptador en la ruta elegida……26

Figura29. Creación del business service……..26

Figura30. Ruta del business service…….26

Figura31. Visualización del business service……...27

Figura32. Menú del business service……27

Figura33. Pestaña general del business service……27

Figura34. Pestaña transport del business service…….…….28

Figura35. Pestaña JCA transport del business service…….….28

Figura36. Parte de la WSDL elaborada……29

Figura37. XSD generada del adaptador……....29

Figura38. Parte de XSD elaborada……30

Figura39. Parte de XSD elaborada……30

Figura40. Creación de transformaciones……..31

Figura41. Elección de nombre para la transformación…….31

Figura42. Búsqueda del elemento para agregarlo a la transformación…….31

Figura43. Gráfico de mapeo…….32

Figura44. Código Generado por el mapeo……32

Figura45. Gráfico de mapeo del response……33

Figura46. Código generado por el mapeo del response……33

Figura47. Gráfico de mapeo del error……...34

Figura48. Código Generado por el mapeo del error…….34

Figura49. Flujo del proxy services……35

Figura50. Creación del proxy services……..35

Figura51. Pestaña general del proxy services……35

Figura52. Grafica del message flow del proxy services……36

Figura53. Grafica de configuración del routing……36

Figura54. Grafica de configuración del assign…….37

Figura55. Grafica de configuración del replace…….……...37

Figura56. Grafica de configuración del publish……38

Figura57. Grafica del flujo del error…….……38

Figura58. Generación del jar…….39

Figura59. Ruta del jar……....39

Figura60. Ingreso a la consola del OSB……39

Figura61. Ingreso al centro de cambios…….40

Figura62. Ingreso a la administración del sistema…….41

Figura63. Búsqueda del jar……....42

Figura64. Elementos del jar……...42

Figura65. Grafica de activación de los cambios……42

Figura66. Validación de cambios……..42

Figura67. Menú de SOAPUI5.5.0…….43

Figura68. Ventana con la WSDL importada…….43

Figura69. Resquest importado por la WSDL……43

Figura70. Request del error de estructura…….44

Figura71. Response del error de estructura……..44

Figura72. Request del caso exitoso…….……..44

Figura73. Response del caso exitoso…….44

Figura74. Request del error de negocio…….45

Figura75. Response del error de negocio……..45

Figura76. Request del error técnico……...45

Figura77. Response del error técnico……46

Figura78. Página de inicio del formato de pruebas……...47

Figura79. Plan de pruebas del documento de pruebas unitarias……47

Figura80. Documento de especificación técnica……48

Figura81. Solicitud de peer review en la herramienta Jira…….49

Figura82. Errores encontrados por la persona que realiza el peer review…….49

Figura83. Solicitud de entrega al equipo de QA……50

Figura84. Especificación de la entrega…….…….50

Capítulo 1
Resumen

Este documento tiene como finalidad presentar el trabajo realizado en la práctica profesional en IBM (International Bussines Machines) COLOMBIA como requisito para obtener el título de ingeniero electrónico; donde se explica las actividades desempeñadas en el área GBS (Global Bussines service), la cual brinda capacitaciones en tecnologías como Java, Angular, React JS, BPM, ODM, bus de integración y metodologías Agiles realizadas por personas expertas en dichas herramientas, con la finalidad de desarrollar software bajo estándares de calidad, buenas prácticas, satisfaciendo las necesidades del cliente, por parte del pasante.

En este documento se puntualiza sobre la implementación de servicios de integración web a través del Oracle Service Bus (OSB), con el fin de cumplir las necesidades de Seguros Bolívar. Generando adaptadores de base de datos para la conexión al OSB, creación de wsdl y xsd con los campos requeridos para el servicio. Así como elaborando documentación técnica para el cliente, buscando desplegar en la consola de Oracle para realizar pruebas unitarias. De igual forma se trabajó bajo la metodología agile scrum, usando para tal fin las herramientas Git Hub y Jira, todo esto bajo la supervisión de líderes técnicos, los cuales brindaron capacitaciones y reuniones de trabajo permanentes para efectos de establecer indicadores, posibles errores y pasos a seguir para concluir de manera exitosa él proyecto.

En el desarrollo de este proyecto se observó las ventajas obtenidas al usar un bus de integración de servicios, entre las que se resaltan: un entorno de poca codificación el cual hace más amigable el desarrollo del servicio, una mayor rapidez de implementación que genera un valor agregado al negocio, código de backend más corto, y fácil detección de errores. Por otro lado, la metodología agile scrum permite tener un mayor control del tiempo de trabajo debido a que se usa la herramienta jira para tal fin, la cual permite ingresar el tiempo gastado por cada actividad desarrollada, además se tiene un indicador que visualiza el estado actual de la tarea asignada, los cuales son: In progress, To do, Cancelled, Done, Blocked, testing y Awaiting feedback.

Capítulo 2
Introducción

IBM Colombia en su área GBS presta servicios de desarrollo web para diversos clientes, para lo cual se define roles entre los que resaltan: Project Management, que tiene como función principal recibir el requerimiento del cliente, por consiguiente crea subtareas con el fin de lograr organizar equipos de trabajo, que se dividen en tres frentes que son: Front End, Backend e integración de servicios web; lo que conlleva a hablar de otro rol que es el líder técnico, este es el especialista en alguno de los frentes anteriormente mencionados, elige a los desarrolladores para que trabajen en el requerimiento, habla con el cliente para solucionar bloqueantes que se puedan presentar en la ejecución del proyecto. Otro rol importante es el de los desarrolladores de software que su principal función es generar los algoritmos necesarios para cumplir con el requerimiento solicitado por el cliente.

La implementación del proyecto se enfoca en el desarrollo de un servicio de integración web para Seguros Bolívar, que se divide en siete etapas que son: instalación de software necesario, recibimiento del requerimiento en la herramienta jira, desarrollo del algoritmo, realización de pruebas unitarias, generación de documentación, correcciones de peer review y entrega del requerimiento al equipo de calidad de IBM. Para lo anterior se realizaron capacitaciones con la finalidad de entender el proceso de desarrollo de software según los requerimientos del cliente.

Todo el desarrollo de software de GBS se trabaja bajo la metodología agile Scrum, la cual se basa en la planificación de pequeños bloques o Sprints, en los cuales se realizan pequeñas entregas funcionales del producto al cliente; de igual forma diariamente se realizan reuniones de máximo 15 minutos donde todos los miembros del equipo responden las siguientes preguntas; ¿Qué se hizo ayer?, ¿Qué se va a hacer hoy?, ¿Qué bloqueantes tienen?, y a partir de estas preguntas se resuelvan interrogantes del equipo, todo esto con el fin de tener una planeación diaria y ordenada de trabajo. Los roles de la metodología scrum son semejantes a los cargos de IBM, lo es, por ejemplo: en Scrum se tiene el rol Project Owner y su equivalencia en la organización es el cargo Project Management(PM), otro ejemplo es el cargo de líder técnico que es semejante al rol de Scrum máster, y el equipo de desarrolladores es el mismo tanto para Scrum como para IBM; el control de los proyectos en curso se realiza mediante el aplicativo jira, este permite crear y asignar las actividades que se necesitan, tiene varios estados para las tareas los cuales son: In progress, To do, Cancelled, Done, Blocked, testing y Awaiting feedback, esta se relaciona con las preguntas que se realizan en la metodología scrum. Desarrollar software con la herramienta jira permite darle un valor agregado a la gestión del trabajo asignada por IBM, debido al mayor control de tiempos en los proyectos desarrollados.

Capítulo 3
Problema

Seguros Bolívar es una empresa que crece en sus distintas líneas de negocio, lo que permite que varias de estas brinden atención por medio de la virtualidad, por lo tanto en la forma tradicional de trabajo se observa como el equipo de negocio construye los requerimientos y los solicita al equipo de desarrollo, esto genera el riesgo que no exista una buena interpretación de las actividades a realizar, lo que conlleva a que los objetivos del producto y la infraestructura tecnológica no estén coordinados, por consiguiente se pueden presentar reprocesos y retrabajos afectando el costo del proyecto y la oportunidad de mercado. Por otro lado, si el usuario final realiza una solicitud, se presentarán retrasos debido al gran flujo de datos que maneja la plataforma virtual, y si es necesario agregar una nueva línea de negocio al portal de oficina virtual de Seguros Bolívar, resulta ineficiente debido a un mayor esfuerzo, trabajo, recurso económico y tiempo para desarrollar la integración.

Es importante tener en cuenta que para ejecutar la inclusión de los nuevos negocios a la oficina virtual de Seguros Bolívar, como por ejemplo en el negocio de seguro de vida de Liberty, se genera un problema porque los datos de esta nueva adquisición se encuentran en los servidores de Liberty, por consiguiente para realizar el traslado de la información en alguna arquitectura tales como: capas, cliente servidor y distribución modular, se obtiene un código extenso de poca eficiencia y con alta probabilidad de error a nivel de backend; además que a nivel de conexión puede que alguna de las partes (backend o base de datos) no soporte la comunicación, por lo que se tendría que realizar una nueva codificación para los productos nuevos pese a que manejan la misma base de datos y, en caso de llegar a presentarse un control de cambios sería bastante difícil realizarlo debido a que se tendría que modificar varios componentes de la aplicación; lo que conlleva a que el costo de mantenimiento y operatividad de infraestructura tecnológica sea bastante grande, asimismo, no permite que las personas que trabajan para esta nueva asesoría la puedan realizar.

Capítulo 4
Justificación de la creación de servicio web de integración

La arquitectura orientada a servicios (SOA), busca la integración de los objetivos del negocio con la infraestructura de tecnología, a través de la implementación de servicios, con el fin de responder fácil y rápidamente a controles de cambios. Las empresas están poniendo en marcha SOA debido a las múltiples ventajas que ofrecen entre las que se resalta: enfilar y acercar las áreas de tecnología y negocio, se minimiza la perdida de datos, permite el desarrollo de la aplicación en menor tiempo, reducción de carga de la infraestructura, facilita la innovación y la adaptación. Otro factor importante de la arquitectura SOA son los actores que son: consumidor del servicio, proveedor del servicio y registro de servicios, para este caso se hablara de la conexión del consumidor y del proveedor del servicio, debido a que el consumidor es el backend del proyecto que desarrolla IBM y el proveedor es la base de datos a la cual se tiene acceso, con respecto al registro de servicios no se tiene acceso a esta información en razón a que se maneja directamente por Seguros Bolívar.[2]

El proveedor de servicios para este proyecto no tiene interacción directamente con el consumidor, sino a través de un bus empresarial de servicios (ESB) que hace referencia a un componente de software que realiza integraciones de backend que permite que estas sean reutilizables para nuevas aplicaciones, entre las características que más resaltan son: alta disponibilidad debido a que si se presentan fallas no son propias del ESB, tiene un conjunto amplio de adaptadores para permitir conectar sistemas que no manejan el mismo protocolo de comunicación, baja codificación debido a que proporciona un entorno visual en el que se pueden realizar algunas tareas necesarias como lo son transformaciones de parámetros y flujo de proxy services, interprete multiformato debido a que logra entender varios lenguajes y lo traduce para parte emisora y receptora, e implementar auditorias de las peticiones que se realizan, asi como el control de errores debido a que por cada backend se presente un error diferente, y genera validación de campos de entrada.[3]

El ESB es un componente importante en SOA debido a que permite que los diferentes servicios interactúen entre sí, sin tener una comunicación directa entre ellos utilizando protocolos de red como lo son SOAP (protocolo simple de acceso a objetos), HTTP O JSON, estos servicios se pueden crear desde cero pero genera problemas debido a que la codificación es bastante robusta, y lograr la integración puede volverse una tarea difícil debido a que se deben hacer traducciones e interpretaciones a cada backend para que entienda el otro, y genera problemas de mantenimiento además no sigue los principios de SOA de eficiencia.[4]

Capítulo 5
Objetivos

General

Implementar servicios web de integración usando el OSB, para satisfacer las necesidades de Seguros Bolívar

Específicos

- Crear adaptador que permite conectar la base de datos del requerimiento con el Oracle Service bus (OSB).
- Implementar el despliegue del servicio en la consola del OSB
- Implementar pruebas unitarias al servicio, para asegurar un funcionamiento óptimo para la posterior integración con los demás equipos
- Generar el documento de guía de instalación y especificaciones técnicas, para que el cliente tenga una referencia de los procedimientos de funcionamiento.

Capítulo 6
Marco Referencial

Marco Teórico

En el año 2002 Grartner y Roy Schulte, plantean uno de los primeros conceptos de "Enterprise Service Bus", que definía una categoría de software presente en el mercado, pero a lo largo de los años no se tiene un concepto fijo, ni lo que debe ofrecer el ESB, el concepto varía dependiendo del fabricante como, por ejemplo:

Para kress, Mair y Normann un bus de integración permite variedad de conexiones punto a punto entre usuarios y servicios proveedores como lo define la arquitectura del ESB, como se observa en la figura1. (Kress, Maier & Normann, 2013).

Figura 1. El patrón de arquitectura ESB (Kress, Maier & Normann, 2013)
(Kress, Maier & Normann, 2013)

El ESB es tipo de componente de software que realiza integraciones (traducción de modelos de datos, conectividad, enrutamiento y solicitudes) con el fin que se encuentren disponibles como interfaces de servicios las cuales son reutilizables en nuevas aplicaciones. Para implementar el ESB se tiene un tiempo de ejecución de integración especialmente diseñado para la aplicación que la necesite y con las herramientas necesarias para tener una productividad mejor, además el ESB es un componente fundamental en la arquitectura SOA, la cual define el método de la reutilización de software en diferentes servicios, las cuales tiene estándares de comunicación comunes para mayor facilidad a la hora de realizar las conexiones de nuevas aplicaciones, esta es una breve explicación de la definición de bus de integración que tiene IBM("Enterprise Service Bus(E.S.B)", 2019)

Un Enterprise Service Bus, es un software que permite la integración de aplicaciones, en la arquitectura SOA (arquitectura orientada a servicios), la cual maneja diferentes protocolos de transporte de mensajería (HTTP, JMS, FTP, SMTP), por lo tanto elimina las conexiones directas (punto a punto) entre el proveedor y el consumidor, ya que siempre

pasan por el ESB, lo que genera un desacople mayor entre cliente y proveedor esto con lleva a darle un valor agregado a los servicios, como se observa el figura2 (Safelayer Marketing D, 2016).

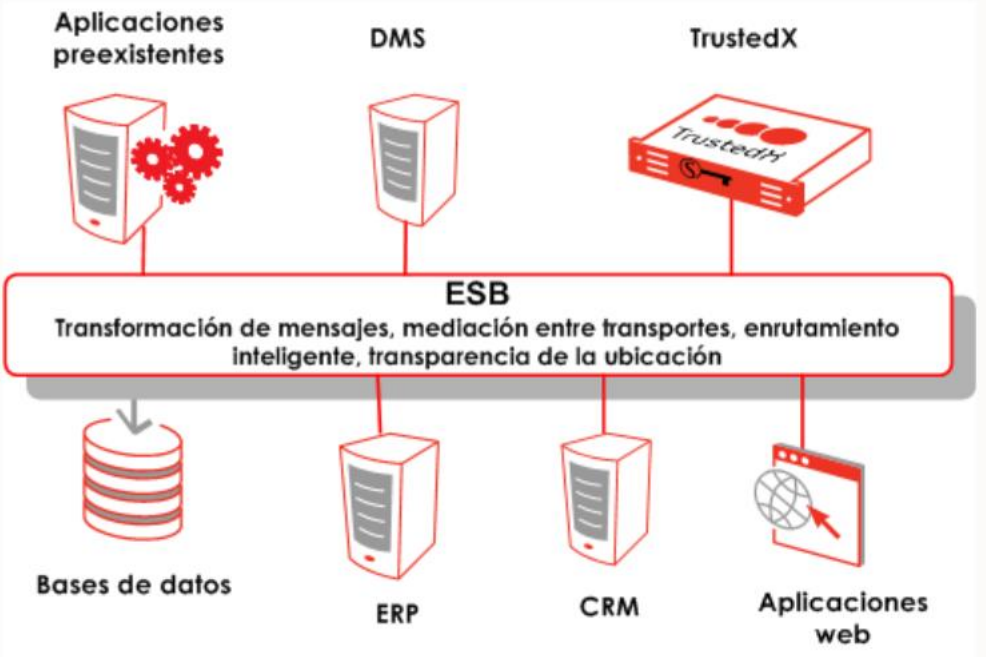

Figura 2. Integración de sistemas utilizando las capacidades de un ESB (Safelayer Marketing D, 2016).

Arquitectura SOA

La arquitectura SOA nace de la evolución de la arquitectura orientada a objetos hasta llegar a la creación de servicios, la cual tiene incorporaciones de software para coordinar procesos de negocios, por ejemplo: los primeros sistemas de software nacen para la automatización de la planeación en la manufactura, su nombre era Material Requirement Planning, el éxito de estos sistemas planteo la necesidad de que fueran útiles para toda la empresa, en caso de los que se dedicaban a servicios se generó la necesidad de construir modelos funcionales para automatizar la gestión, lo que con lleva a que las empresas generen su propio software especializado de diferentes marcas y resultaba incompatibles para la transferencia de datos, gracias a esto surge la necesidad de integración de datos entre aplicaciones conocida como Enterprise Application Integration, esto genera la creación de la arquitectura orientada a servicios (Mario Guerrero, 2018)

SOA representa una etapa importante en el desarrollo de software, debido a que antes que existiera la conexión de una aplicación a datos o funcionalidades alojados en otro sistema, requería una integración bastante compleja punto a punto, tanto que los desarrolladores

tenían que volver a hacer el proceso por cada nueva aplicación, mientras que trabajando con SOA no se llega a un nivel de integración profunda.

Una arquitectura orientada a servicios define que los componentes que se utilizan sean reutilizables a través de interfaces de servicios, las cuales usan estándares de comunicación bastante comunes para poder conectarse rápidamente con nuevas aplicaciones, sin tener que realizar una integración muy profunda.

Una gran ventaja que tiene las interfaces de servicios es que su acoplamiento es bastante flexible, lo que conlleva que pueda llamarse con poco o ningún conocimiento de cómo se realiza la integración. los servicios se exponen utilizando protocolos de red estándar como, por ejemplo: SOAP (protocolo simple de acceso a objetos), HTTP O JSON, para enviar solicitudes para leer o cambiar datos. Otra ventaja que tienen es que se publican de una manera rápida lo que permite que los desarrolladores los encuentre rápidamente y los pueda reutilizar para otras aplicaciones. En el siguiente esquema se presenta la comparación entre SOA y sistemas monolíticos.

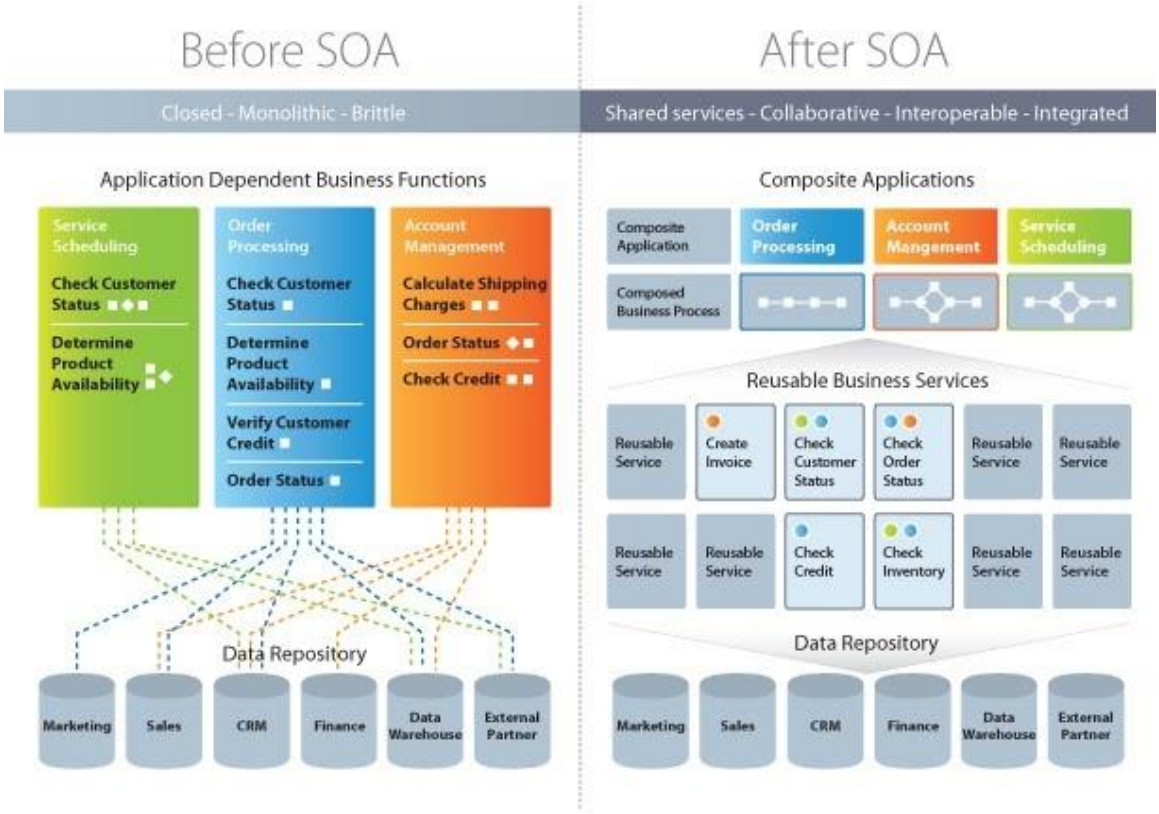

Figura 3. Se presenta la comparación entre SOA y sistemas monolíticos ("Arquitectura Orientada a Servicios (SOA)", 2018)

Como se puede ver, SOA se contrapone a las aplicaciones monolíticas y propone la implementación de servicios que se pueden combinar para generar las funcionalidades necesarias.
Por tanto, una arquitectura SOA propone una arquitectura desacoplada (o con acoplamiento débil) que tiene los siguientes beneficios:

- Aumenta la interoperabilidad intrínseca.
- Incrementa la federación.
- Aumenta las opciones de trabajar con diferentes proveedores.
- Aumenta la alineación entre negocios y tecnología
- Aumenta el retorno de la inversión.
- Aumenta la agilidad organizacional.
- Reduce la carga operativa de TI. ("Arquitectura Orientada a Servicios (SOA)", 2018)

Metodología Scrum

La metodología scrum hace parte de las metodologías agile para el desarrollo de software, la cual tiene de fundamento construir la funcionalidad de un producto de mayor valor para el cliente, basado en los principios de inspección, adaptación, autogestión e innovación, y su principal objetivo es máximizar el retorno de inversión.El cliente con la metodología scrum se entusiasma con el proyecto debido a que puede observar cómo avanza el requerimiento, y también le permite hacer cambios al software en el inicio de cada ciclo.

Roles en el Scrum:

- **Product Owner:** Hace referencia al "dueño del producto", este tiene como funciones transmitir los requerimientos del cliente, priorizar tareas, arma el equipo de trabajo, crea las historias de usuario, habla con él cliente.
- **Scrum Máster:** Es la persona al mando del proyecto se encarga de liderar al equipo de desarrolladores para cumplir con las tareas y tiempos, también una de sus funciones es eliminar bloqueos

- **Development Team Members:** Son las personas encargadas de desarrollar el proyecto.

El proceso Scrum:

- **Listado de requisitos:** El cliente presente los objetivos y/o requisitos necesarios para el proyecto, teniendo en cuenta las prioridades para la posterior agrupación por iteraciones y entregas
- **Planificación:** Cuando ya se tiene los requerimientos, el product owner se encarga de realizar una estimación de fuerza para saber el tiempo de entrega, y elegir el equipo
- **Reunión de sincronización:** Trata de una reunión diaria del equipo de trabajo, la cual tiene una duración máxima de 15 minutos, se realiza en el mismo lugar a la misma hora todos los días en la cual se responden 3 preguntas
  - ¿Qué se hizo?
  - ¿Qué se hará?
  - ¿Qué dudas o problemas se presentaron?
- **Revisión del Sprint:** En este el equipo de trabajo realiza una reunión con el cliente donde se presente el trabajo que se realizó en el sprint, después de esto el cliente da su opinión y expresa adaptaciones o cambios necesarios
- **Revisión retrospectiva:** Esta es una reunión de todo el equipo de trabajo para realizar un análisis de como trabajaron y aspectos a mejorar.

A continuación, se muestra la representación gráfica del proceso scrum ágil.

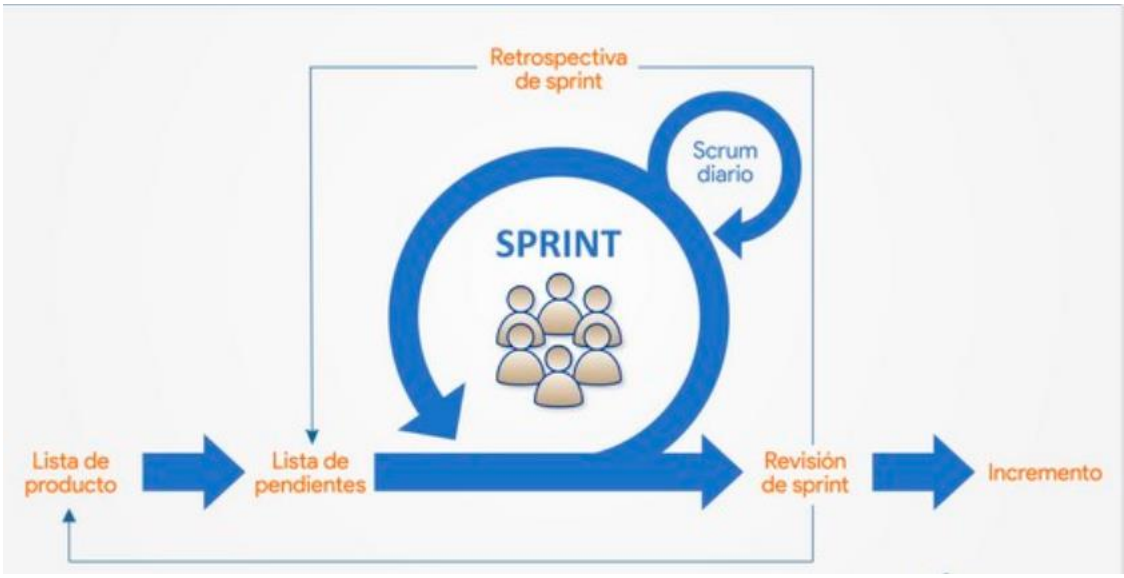

Figura 4. Representación del proceso en la metodología Scrum agile(“Metodología Scrum”,2020)

Beneficios:

- **Cumplimento de expectativas:** El cliente desde un inicio establece expectativas del proyecto, las cuales se dividen en historias de usuario con el product owner y establece prioridades de trabajo.
- **Flexibilidad a cambios:** Tiene una gran ventaja con respecto al cambio debido a que, gracias a el ciclo de iteraciones, no produce gran impacto cambiar lo que decida el cliente.
- **Reducción del Time to Market:** El cliente en cada iteración observa una parte del producto funcional lo cual permite que pueda hacer uso de él, sin estar terminado.
- **Mayor calidad del software:** Debido a que al final de cada iteración se debe tener una parte funcional de software, permite que se tenga una calidad de software mejor
- **Mayor productividad:** Se consigue debido, a la autonomía que se le da al equipo de poder ejecutor los requerimientos
- **Maximiza el retorno de la inversión (ROI):** Gracias a esta metodología se le da mayor valor al software, lo que permite maximizar el retorno de inversión.
- **Predicciones de tiempos:** Debido a los cálculos de duración de iteración, se puede llegar a calcular cuando está el producto final terminado, lo que permite una mejor experiencia para el cliente.

Estadísticas del uso de la metodología ágil:

- Mejora en 54% la colaboración entre equipos que no acostumbran a trabajar juntos.
- Aumento del 52% en la calidad del software.
- Aumento del 49% en la satisfacción del cliente con el producto.
- Disminución de un 43% en el tiempo de comercialización
- Reducción de un 42% en el coste de desarrollo

("Metodología SCRUM para desarrollo de software a medida", s.f.)

PL/SQL:

- Es una extensión de programación procedimental del SQL
- Procedural Language/Structured Query Language
- Es un lenguaje de programación incrustado en Oracle.
- Lanzamiento inicial: 1992
- PL/SQL ofrece características modernas de ingeniería de software, tales como la encapsulación de datos, manejo de excepciones, ocultación de información y orientación a objetos.

Arquitectura (Figura 5):

- Se compone de código PLSQL+ sentencias SQL.
- el código PLSQL es ejecutado en un engine llamado PLSQL y las secciones que son sentencias SQL son ejecutadas en el SQL Statement Executor (Oracle Database Server).

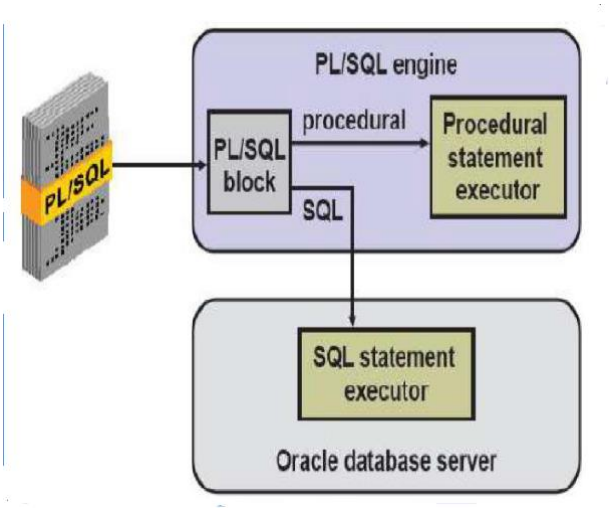

Figura 5. Arquitectura PL/SQL, (Oracle Database Server,2018).

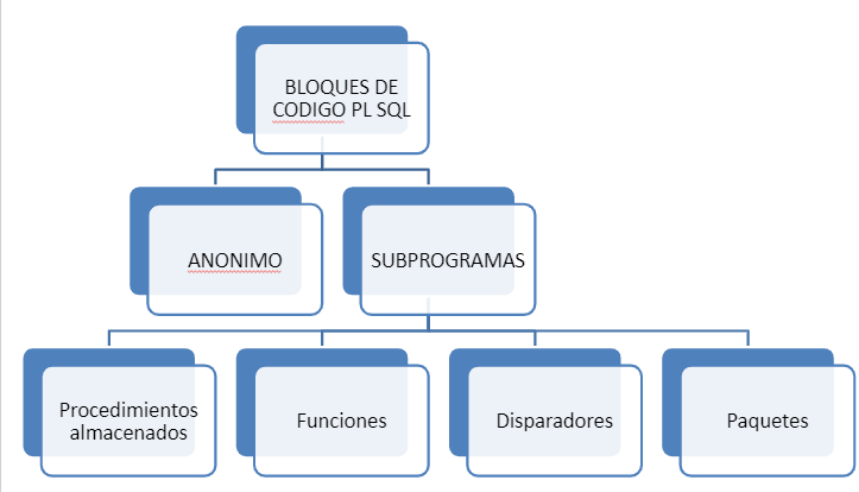

Figura6. Bloques PL/SQL (PLSQL,Henry Alfonso Guio)

STORED PROCEDURE:

Son conjuntos de sentencias en lenguaje SQL con nombre asignado, que se almacenan en bases de datos relacionadas como un grupo, lo que permite ser utilizadas por más programas, estos no están vinculados a una base de datos específica, aunque pueden acceder y modificar datos de esta.

Un procedimiento almacenado proporciona seguridad debido a que la interfaz del usuario y la base de datos solo permite que el primero ingrese o cambie datos, pero no llegan al código del SP, una gran ventaja es que mejora la productividad porque el procedimiento solo se escribe una vez, no cada vez que se ingresen o cambien datos (**w3schools**.com, SQL Procedimientos almacenados de SQL server)

Git Hub:

Fue creado en el año 2005 por Linus Torvalds, es un sitio web para gestión de proyectos, control de cambio, y control de versión de código.

Elementos básicos de un proyecto en Git:

1. **Repositorio:** Es un directorio en git donde se almacenan los archivos del proyecto, para que todos los colaboradores tengan acceso a él.

2. **Branch o rama:** Copia del repositorio central, donde se puede trabajar de forma aislada sin afectar a las demás personas que están trabajando, se permite fusionar con otras ramas, y con el repositorio central.

JIRA:

Es software diseñado para ayudar a la gestión de equipos, debido a que se puede planificar sprints, hacer seguimiento a incidencias, registrar estimación y trabajo, informes de progreso, tableros scrum, distribuir tareas y crear historias de usuario e incidencias, en relación con lo anterior permite, trabajar bajo la metodología scrum.

WSDL (Web Service Description Language):

Significa Web Service Description Language, es un protocolo basado en XML que permite describe íntegramente servicios web, se usa para generar una entrada al servicio web. Dentro de la WSDL se describe:

- Tipos de datos: se ingresa la dirección del esquema que tiene los datos a definir.
- Mensajes: se define los elementos que tiene el mensaje que se va a enviar para este caso es el request y el response.
- Tipos de puerto: se define las operaciones permitidas y los mensajes que se van a intercambios request y response.
- Bindings: Se define el protocolo de comunicación.
- Servicios: Conjunto de puertos y direcciones del consumo de servicio.

Para poder consumir una WSDL se debe crear un Cliente Consumidor, que pueda interpretar la información que trae la WSDL, y se pueda conectar a esta para poder realizar el consumo correcto.

XML:

Significa Extensible Markup Language, proporciona una plataforma para definir elementos dentro un formato personalizado permite definir otro lenguaje marcado.

SERVICIOS WEB:

Es un sistema de software que permite la comunicación de máquina a máquina por medio de una red, realiza un conjunto de tareas descritas en una WSDL, la implementación de servicios web contiene acoplamientos flexibles y orientados a componentes. [4]

XSD:

Significa XML schema definition, permite la definición de estructura de un documento XML, así mismo su validez.

Capítulo 7
Diseño Metodológico

Para la realización de este trabajo se realiza bajo la metodología AGILE SCRUM, por lo tanto, se trabajó de la siguiente manera

1. El Scrum Máster arquitecto de software Jesús Graterol, toma el requerimiento que necesita seguros Bolívar para transmitirlo al líder técnico, este último se encarga de elegir el equipo el cual va a trabajar para el requerimiento.

2. Luego de tener conocimiento del requerimiento el líder lo divide en historias de usuario, lo que quiere decir que se divide en tareas más pequeñas para que el equipo trabaje sobre esto y sepa cuales tiene más prioridad que otras.

3. Después de dividir el trabajo en los diferentes equipos que son los siguientes: FRONTEND, BACKEND e INTEGRACIÓN.

4. Se asignan las tareas principales de cada equipo.

5. En el caso de integración se entrega la información de la base de datos (accesos, pool de conexiones, nombre del paquete y nombre del procedimiento) a la cual se tiene que dirigir el servicio de bus para traer la información.

6. Creando la conexión a la base de datos por parte del bus se procede a crear el contrato del OSB, que corresponde a wsdl y xsd esto con el fin de entregarlo al equipo de BACKEND para que generen el cliente. (si el contrato sufre alguna modificación se tiene que avisar que fue lo que se cambió para calcular el impacto a nivel de backend o frontend).

7. Después de tener el adaptador de la base de datos y el contrato se procede a crear el flujo, y las transformaciones para el servicio de bus

8. Ahora llega el momento del despliegue, en el cual se genera un jar para subir a la consola del bus.

9. Se realizan pruebas unitarias para probar el servicio y generar la documentación.

10. Se envía un correo al equipo de trabajo para revisión de PeeReview

11. después de la respuesta de este se tiene que corregir los errores para realizar la entrega.

PARTICIPANTES:

1. Santiago Andres Caceres Silva, Practicante quien realiza el proyecto

2. Jorge Sánchez, líder técnico de IBM para Seguros Bolívar

3. Jesús Graterol, Arquitecto de Software de IBM.

4. Johary Pérez, Project Management

RECURSOS DISPONIBLES:

1. Computador de IBM.
2. Conexión a las redes de Seguros Bolívar.
3. Permisos para conexión de base de datos de Seguros Bolívar.

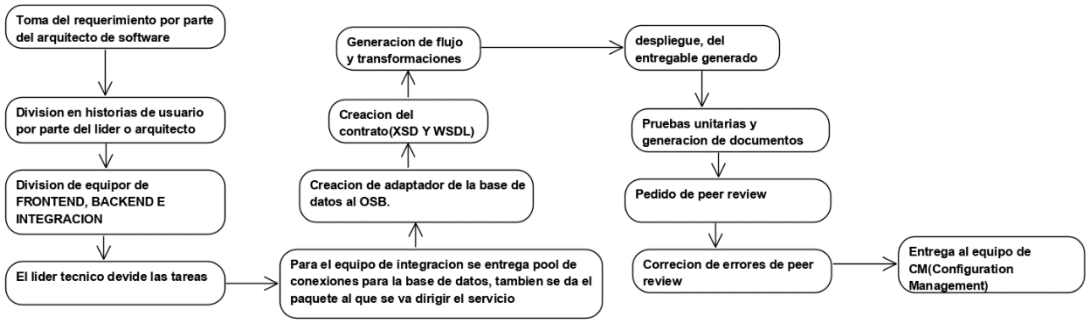

Figura 7. Imagen de la metodología desarrollada (Fuente: Autor).

Capítulo 8
Desarrollo

8.1. Requisitos

- JDK de Java, versión 1.8.0_144 o superior.
- Oracle Service Bus, Eclipse.
- Jdeveloper 11g.
- Sql Developer.
- Check Point Vpn.
- SOAPUI 5.5.0.

Después de tener instalado el ambiente de trabajo el siguiente paso es observar cómo se toma el requerimiento y los pasos para realizarlo, a continuación, se observar cómo se realizó el servicio de OFVConsultaFondoLiberty para Seguro Bolívar todo esto con el fin de tener claridad de cómo se desarrollan servicios de integración web.

8.2. Asignación del Requerimiento en la herramienta Jira:

A continuación, se observará como se adquiere el requerimiento desde la herramienta Jira, el cual debe tener la descripción de la actividad a realizar, las credenciales de acceso a la Base de datos, y el estado de la tarea, el cual modifica el desarrollador a “In progress” cuando lo está haciendo, a “blocked” si presenta algún bloqueo o “Done” si ya se finalizó. (tener en cuenta que por contrato de confidencialidad no se puede observar información completa):

Figura 8. Asignación del requerimiento en jira (Fuente: Autor).

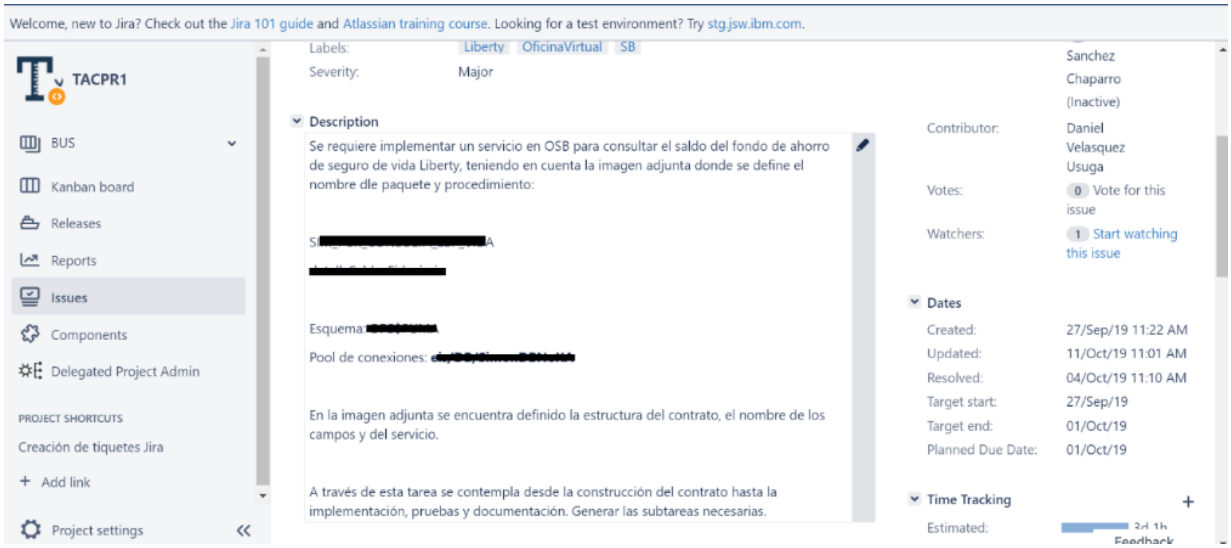

Figura 9. Descripción del requerimiento en jira (Fuente: Autor).

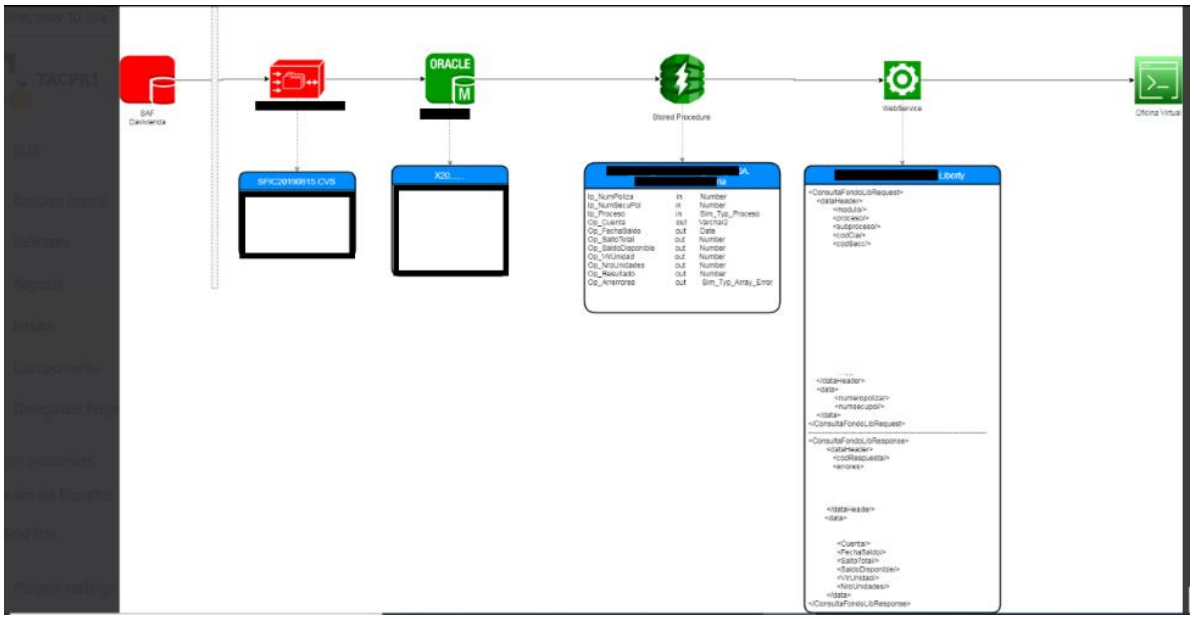

Figura 10. Grafico del requerimiento (Fuente: Seguros Bolivar,2019).

8.3. Creación de servicio de integración:

Se realiza los primeros pasos de configuración de un servicio de integración Web, para lo cual se crea en primer lugar una configuración de OSB, para después crear el proyecto sobre el cual se va a trabajar, esto para que el entorno de desarrollo integrado conocido como eclipse, reconozca las librerías a trabajar y evitar posibles errores por falta de configuración del OSB. Por otro lado, se crean las carpetas necesarias donde se depositarán

los siguientes elementos: Contrato, Bussines, Recursos y proxy, con el fin de obtener un desarrollo donde se evidencien buenas prácticas.

1. Abra el programa Oracle Enterprise Pack for Eclipse.

Figura 11. Icono del OSB (Fuente:Autor).

2. Se crea un nuevo proyecto, entre a ***File – New – Other…***

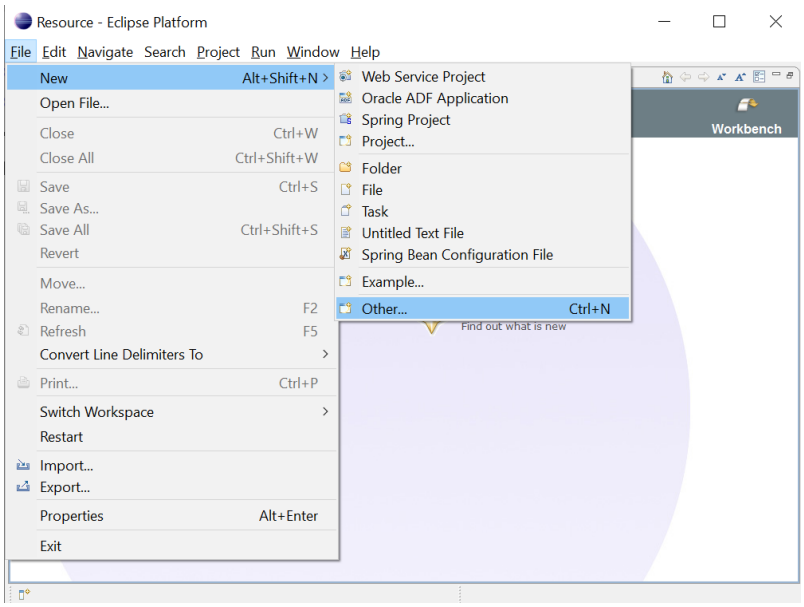

Figura 12. Creación de configuración del OSB (Fuente:Autor).

3. Se busca “Oracle” y se crea un “Oracle Service Bus Configuration Project”

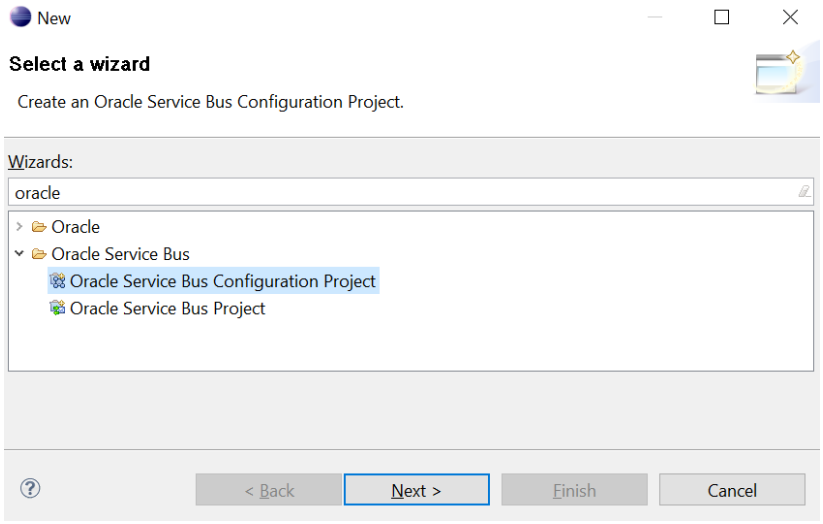

Figura 13. Creación de configuración del OSB (Fuente: Autor).

4. Nombre el proyecto y luego presione el botón ***Finish***.

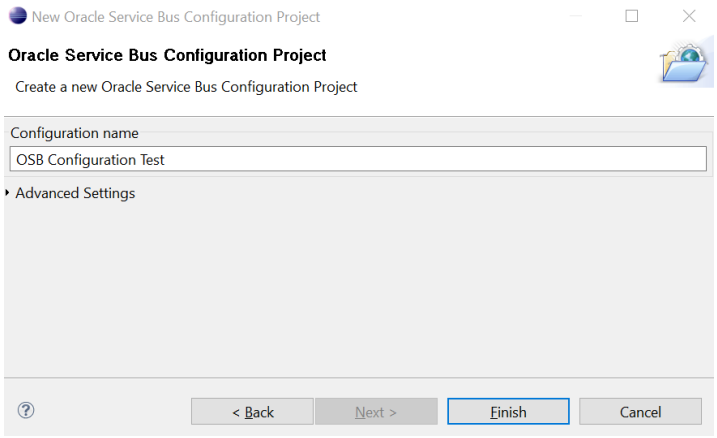

Figura 14. Configuración del OSB (Fuente: Jonathan Sanchez, 2019).

5. Sobre este proyecto se crea un Oracle Service Bus Project

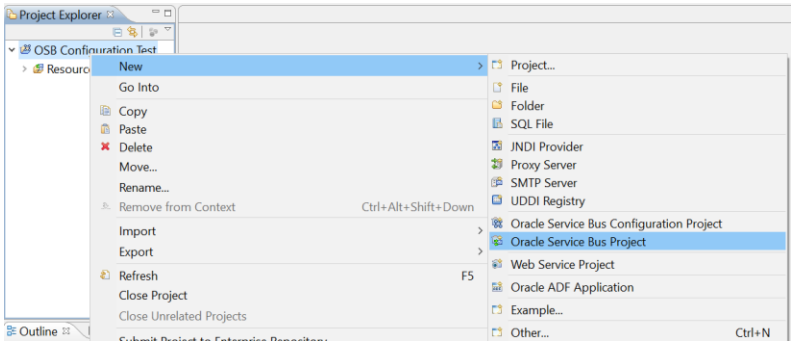

Figura 15. Creación de proyecto del OSB (Fuente: Jonathan Sanchez, 2019).

6. Se nombra el proyecto para este caso se llamará OFVConsultaFondoLiberty y se presiona el botón ***Finish.***

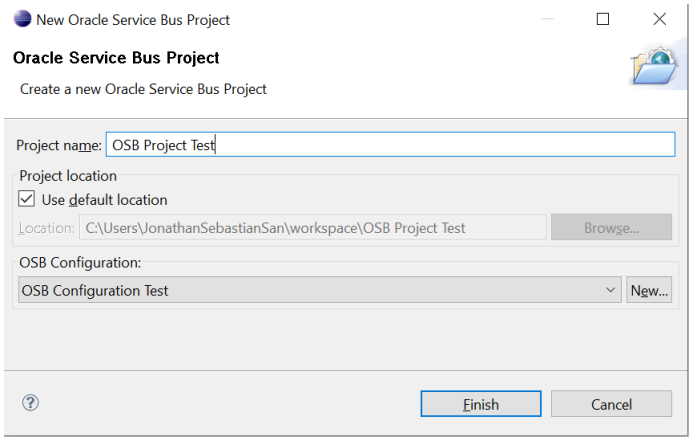

Figura 16. Asignación de nombre del proyecto (Fuente: Jonathan Sanchez, 2019).

7. Se Presiona el botón ***Yes.***

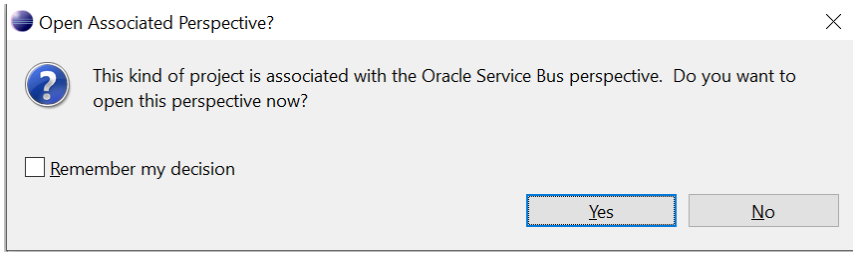

Figura 17. Confirmación de creación. (Fuente: Jonathan Sanchez, 2019).

8. Se crean cuatro carpetas con los nombres de:

- Contrato
- Proxy
- Recursos
- Business

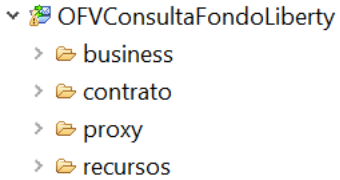

Figura 18. Carpetas creadas (Fuente: Autor).

8.4. Creación de adaptador

La creación del adaptador tiene la finalidad de generar el bussines service y el contrato del stored procedure, para esto se utiliza el programa Jdeveloper el cual permite la conexión entre el stored procedure de la base de datos y el bus de integración mediante el protocolo JDBC. Para esto primero se configura la nueva aplicación en “SOA APLICATION” y se escoge la dirección donde quedara el adaptador, que para el caso se encuentra en la carpeta “recursos” anteriormente creada, después se elige la opción “Database adapter” donde se configura la conexión de la base de datos; Cuando este creado en la carpeta de recursos se genera el business service el cual es el servicio final que expone la base de datos, para este caso es el stored procedure requerido por el cliente.

1. Se Entra a JDeveloper

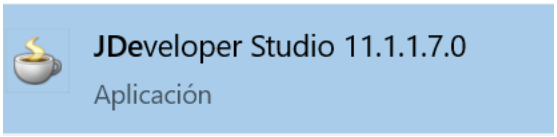

Figura 19. Icono Jdeveloper (Fuente: Autor).

2. Se Realiza click sobre la opción “New Aplication”, se coloca un nombre y elige la opción de “SOA APLICATION” y presiona Next:

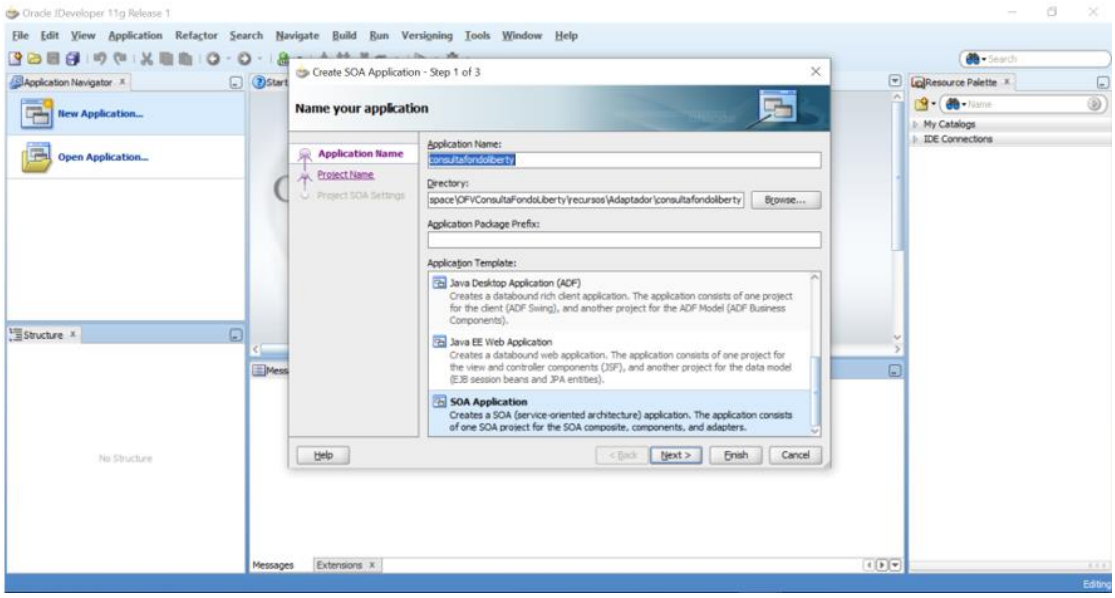

Figura 20. Elección de SOA APLICATION. (Fuente: Autor).

3. Se Coloca un nombre al proyecto, lo siguiente es elegir la ruta que para este caso será la carpeta adaptador creada para la implementación del servicio, los parámetros que están predeterminados no se modifican, y se oprime click en Next.

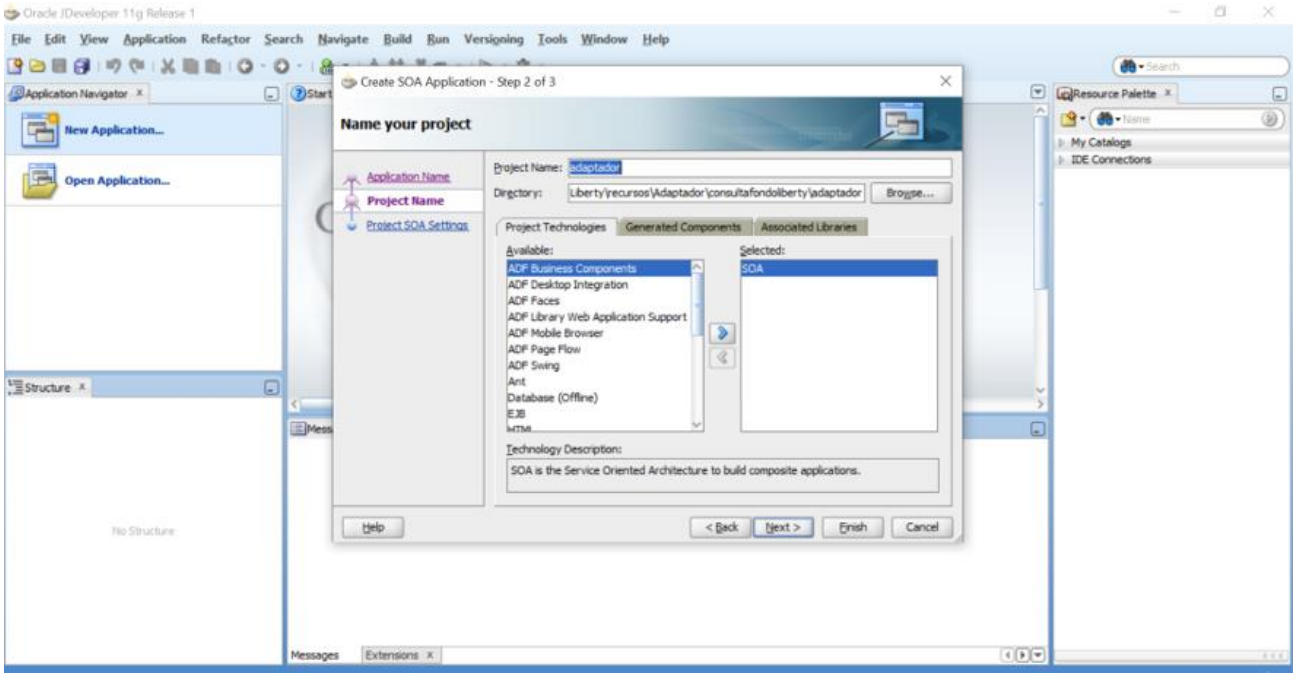

Figura 21. Elección de ruta del adaptador (Fuente: Autor).

4. Se Dejan los valores predeterminados y se oprime click en finish.

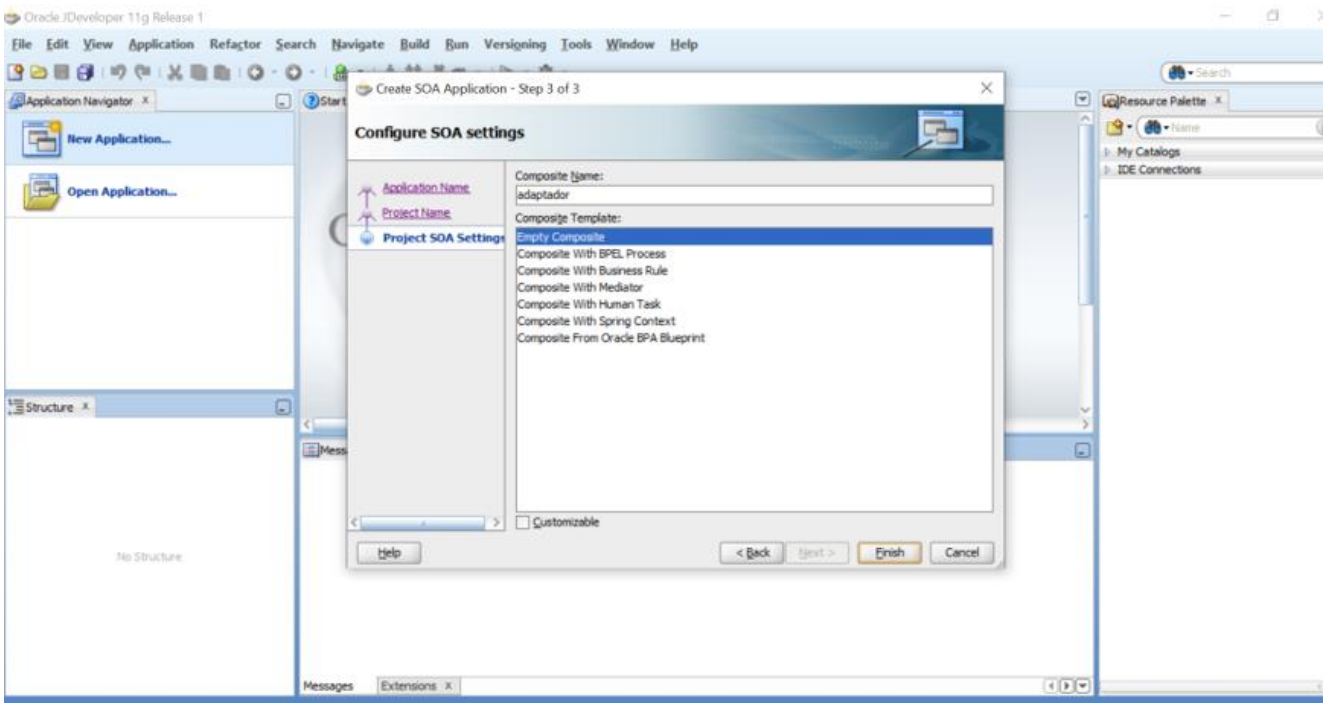

Figura 22. Elección de valores predeterminados. (Fuente: Autor)

5. Después de creado el proyecto se busca la opción de DataBase Adapter, para arrástralo a la pantalla.

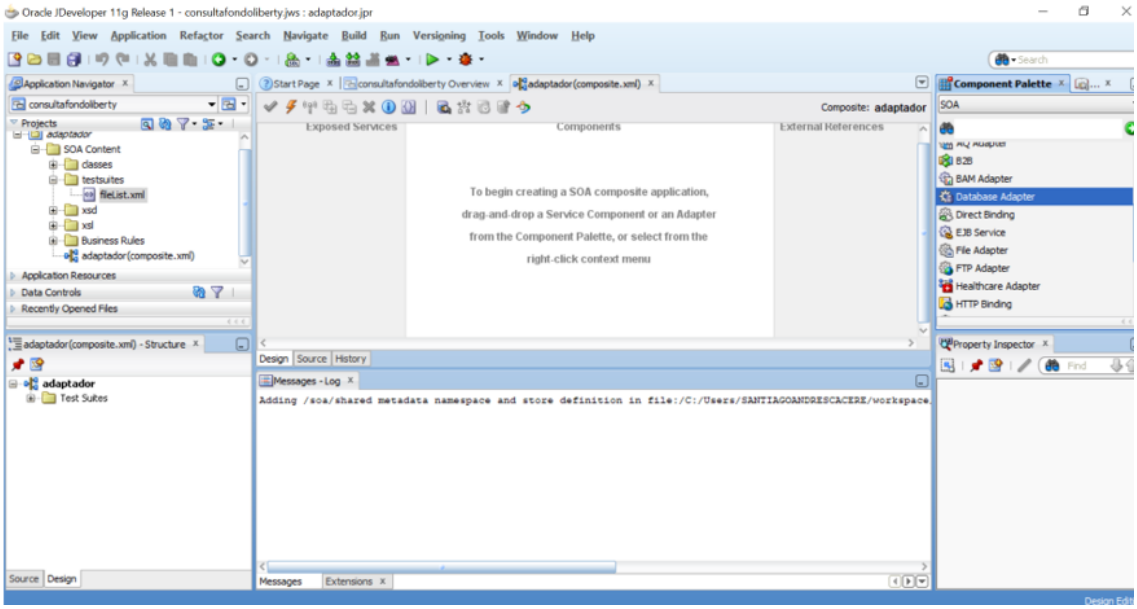

Figura 23. Elección de opción de database adapter (Fuente: Autor).

6. Se desplegará otra pantalla para lo cual se da click, en el "mas" verde de la parte superior derecha:

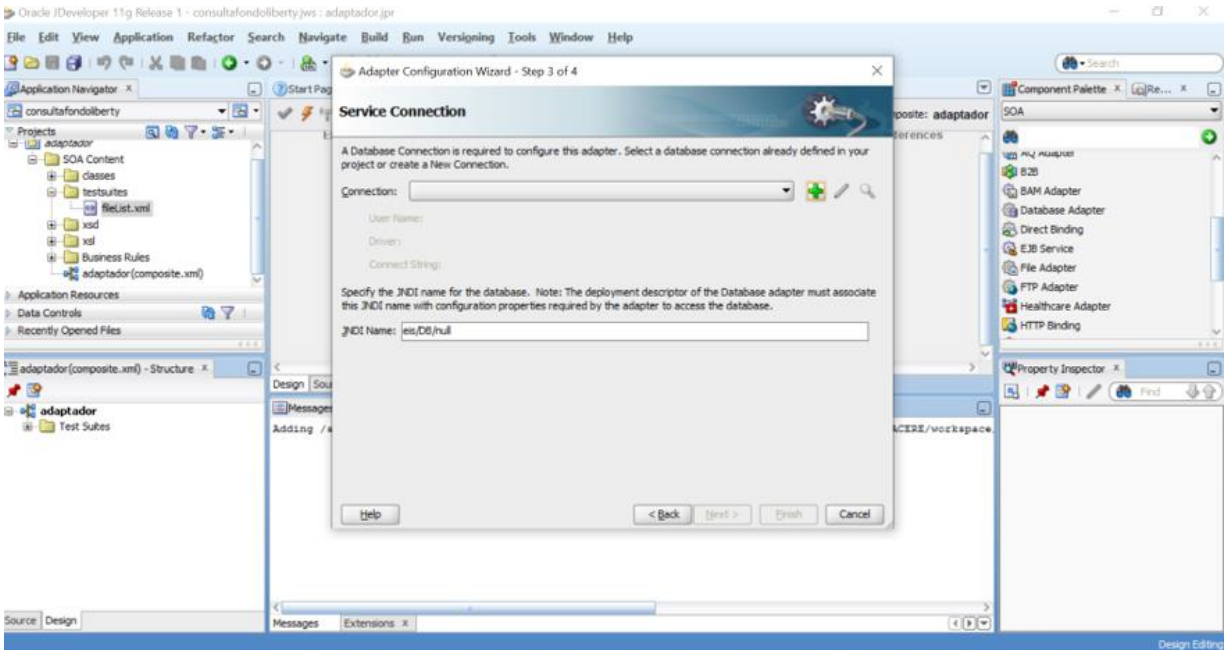

Figura 24. Conexión a base de datos (Fuente: Autor).

7. Se desplegará otra pestaña para ingresar los parámetros para la conexión los cuales son:

- Nombre de conexión: Esta puede ser cualquier nombre debido a que el creador del adaptador es autónomo de establecerlo.
- Nombre de usuario: Hace referencia al nombre de la base de datos.
- Contraseña: Password de la base de datos.
- Nombre del host: Dirección IP de la base de datos
- Puerto: hace referencia al puerto JDBC de la base de datos.
- Nombre del servicio: hace referencia al nombre que le asigna el cliente.

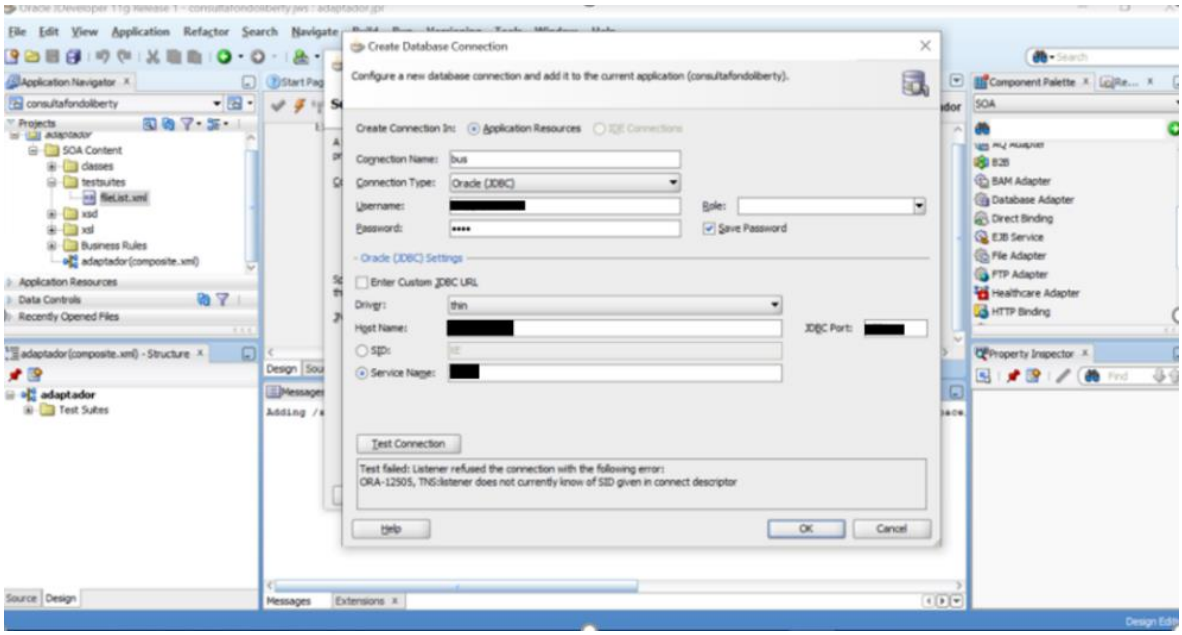

Figura 25. Ingreso de parámetros de base de datos (Fuente: Autor).

8. Después se despliega una ventana y se elige la opción “call stored procedure or function” y se procede a oprimir “Ok”.

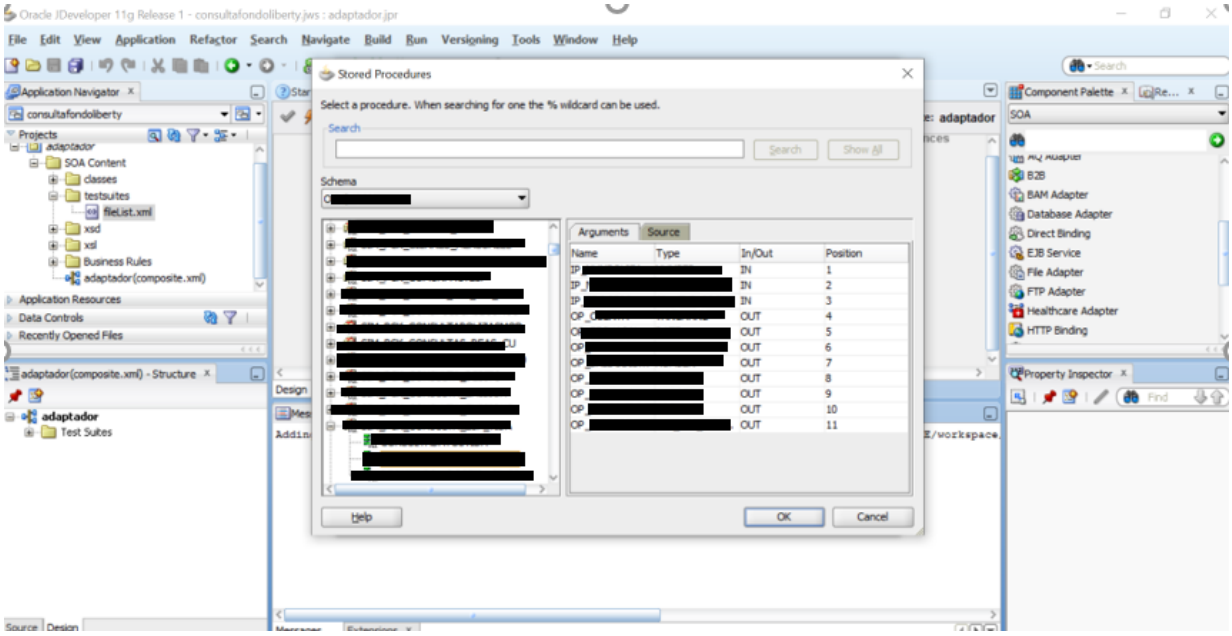

Figura 26. Elección del stored procedure (Fuente: Autor)

9. Ahora aparece una ventana con el stored procedure elegido y se da click en “Next”.

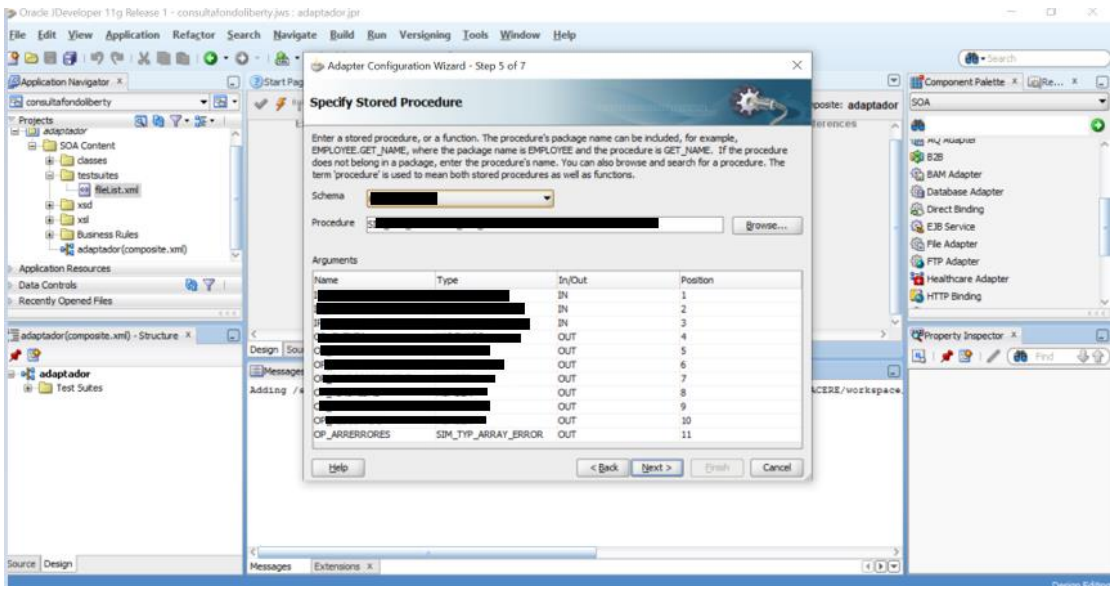

Figura 27. Confirmación del stored procedure escogido (Fuente: Autor).

10. En la ruta que del paso 3, se crea una carpeta que se llama xsd para dejar dentro de este el archivo .xsd que se crea de la generación del adaptador, además se generan archivos .wsdl y.jca.

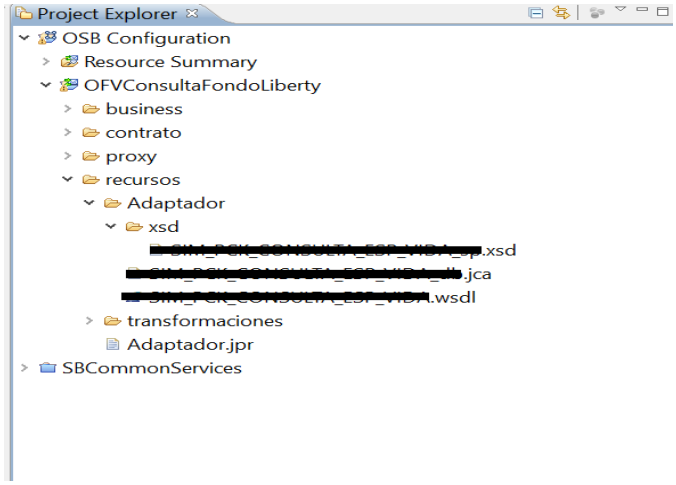

Figura 28. Visualización del adaptador en la ruta (Fuente: Autor).

11. Para el al archivo. Jca, se le da click derecho, y se elige la opción de Oracle service bus, generate service

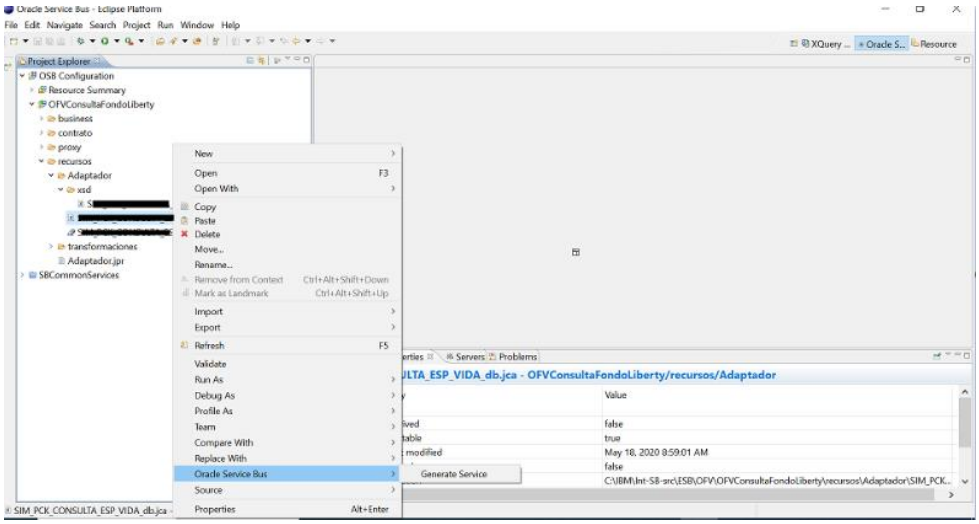

Figura 29. Creación del Business service (Fuente: Autor).

12. Se elige la ruta en la carpeta de business y se oprime ok

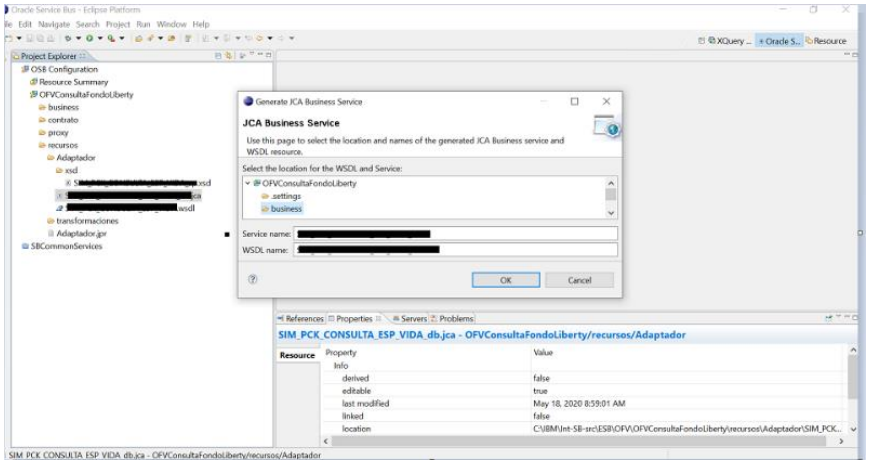

Figura 30. Ruta del Business service (Fuente: Autor).

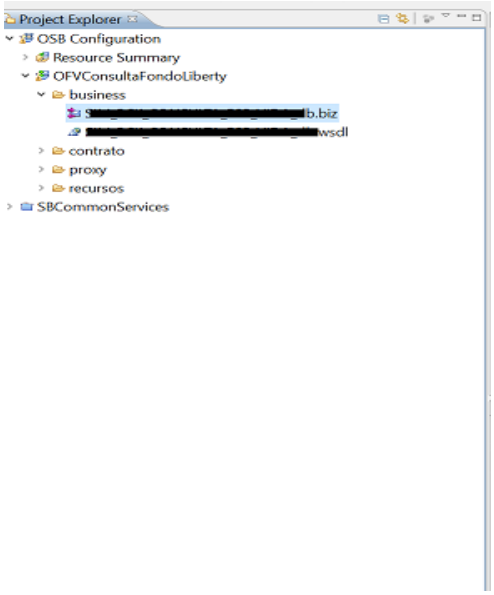

Figura 31. Visualización del Business service (Fuente: Autor).

Acá se observan los archivos que se crean de la generación del servicio, dentro del archivo .biz, genera varias pestañas

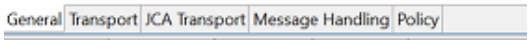

Figura 32. Menu del Business service (Fuente: Autor).

Dentro de la pestaña General, se observa que el servicio que se crea viene de una wsdl.

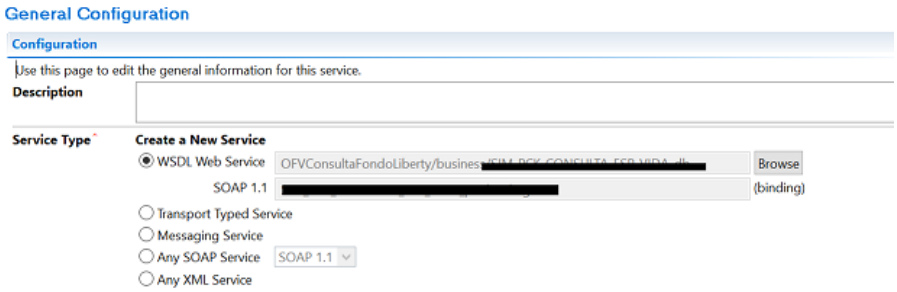

Figura 33. Pestaña General del Business service (Fuente: Autor).

La pestaña Transport nos indica el protocolo que se maneja que en este caso es JCA, el tiempo de respuesta del servicio que por acuerdo con el cliente es de 30 segundos y load balancing algorithm que es round-robin

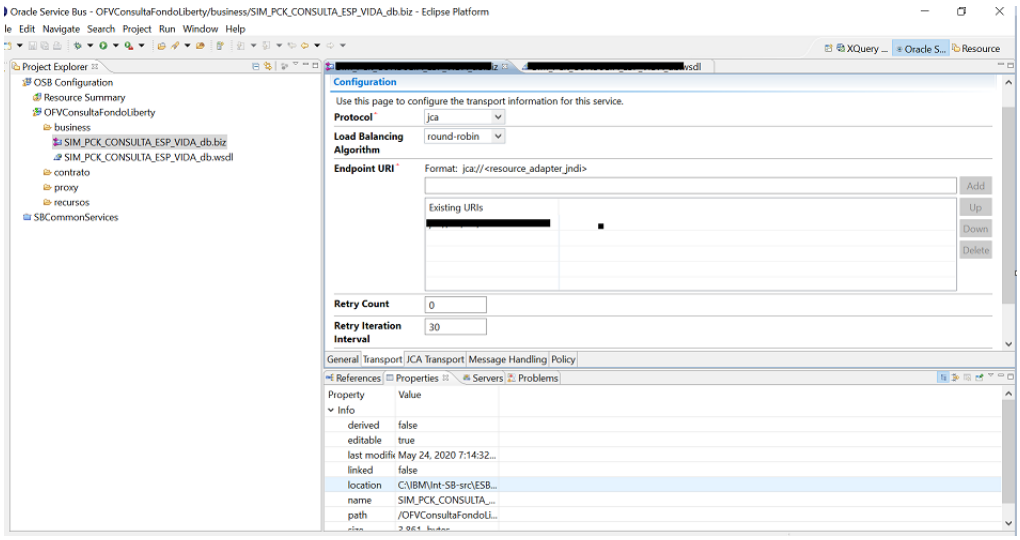

Figura 34. Pestaña Transport del Business service (Fuente: Autor).

La pestaña JCA Transport indica la ruta del archivo JCA, nombre del adaptador, tipo de adaptador y las políticas de despacho

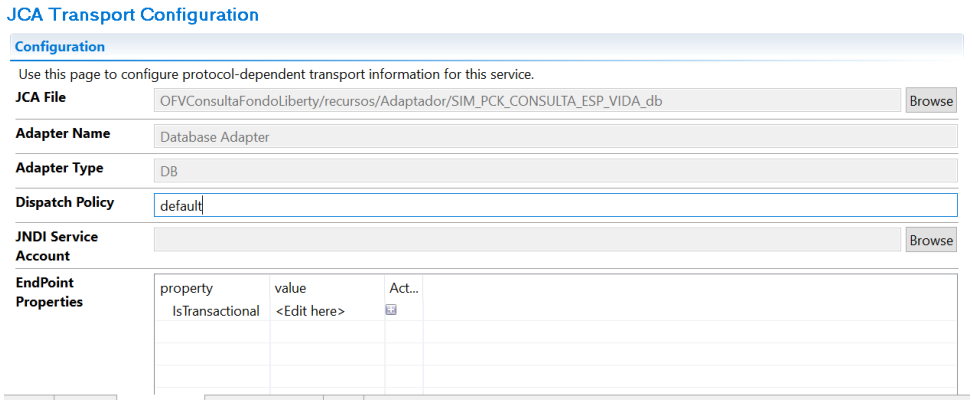

Figura 35. Pestaña JCA Transport del Business service (Fuente: Autor).

8.5. Elaboración del contrato (wsdl y xsd)

La elaboración de la wsdl se debe hacer con precisión debido a que, si se comete un error se presentan problemas graves debido a que esta se le entrega al equipo de backend mediante Jira y estaría trabajando bajo un contrato que presenta fallas, por lo tanto, el proyecto se retrasa y se tendría que se presenta retrabajo.

1. Para la elaboración del script de la wsdl se necesita definir:

- Tipos de datos: se ingresa la dirección del esquema que tiene los datos a definir.
- Mensajes: se define los elementos que tiene el mensaje que se va a enviar para este caso es el request y el response.
- Tipos de puerto: se define las operaciones permitidas y los mensajes que se van a intercambios request y response.
- Bindings: Se define el protocolo de comunicación.
- Servicios: Conjunto de puertos y direcciones del consumo de servicio para este caso, se apunta al ambiente de pruebas de Seguros Bolívar.

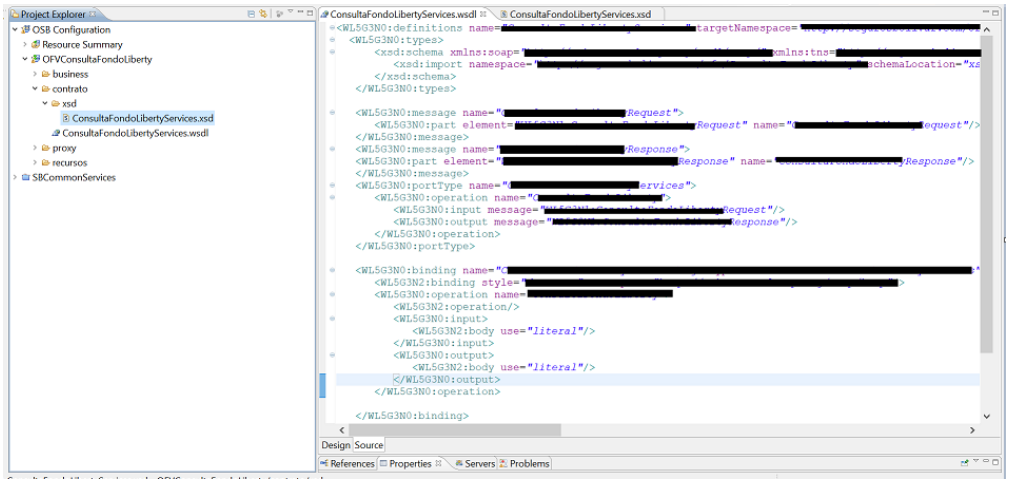

Figura 36. Parte de la WSDL elaborada (Fuente: Autor).

2. Para la definición y elaboración del esquema XSD, se debe tener en cuenta el archivo .xsd que se generó del adaptador, ya que los campos y el tipo de dato deben ser iguales, con el fin que se puedan realizar las transformaciones al OSB. Dentro de este esquema XSD, se establecen las operaciones definidas en la WSDL que para este caso es el request y response, dentro de estas se observaran los campos necesarios con el tipo de dato, y la cardinalidad que permite saber la obligatoriedad del campo, además por estándar de IBM y de Seguros Bolívar dentro del request se agrega una cabecera con campos opcionales a excepción de la versión del servicio, para el response se anexa otra cabecera que solo tenga el campo de "codRespuesta" que indica si el servicio responde correctamente o no, este parámetro esta predeterminado en el procedimiento almacenado de la base de datos, y por último se define un vector de errores el cual establece campos que se usan para los distintos tipos de errores que se presentan.

Figura 37. XSD generada del adaptador (Fuente: Autor).

Figura 38. Parte de XSD elaborada (Fuente: Autor).

Figura 39. Parte de XSD elaborada. (Fuente: Autor).

8.6. Transformaciones o mapeos:

Las transformaciones o mapeos del OSB permite que los campos de la XSD del adaptador se asocien a la XSD que se creó, para que se visualicen en la estructura de esta última y no como lo haría en la base de datos. Para el desarrollo de servicios se crean tres trasformaciones, la primera para el request, la segunda para el response y la tercera para el vector de errores. Este entorno es muy gráfico lo cual lo hace de fácil implementación para el desarrollador. Por otro lado, cuando se realiza el mapeo automáticamente se genera código, por lo general este código no se edita, sin embargo, para este desarrollo se tiene que configurar un condicional IF para el response, debido a que los campos con la información de la base de datos solo se deben mostrar cuando el codRespuesta es cero, de lo contrario se observan los campos del error.

1. Dentro de la carpeta recursos se crea otra carpeta que se llame transformaciones, se oprime click derecho sobre esta, y se busca la opción New XQuery Transformation

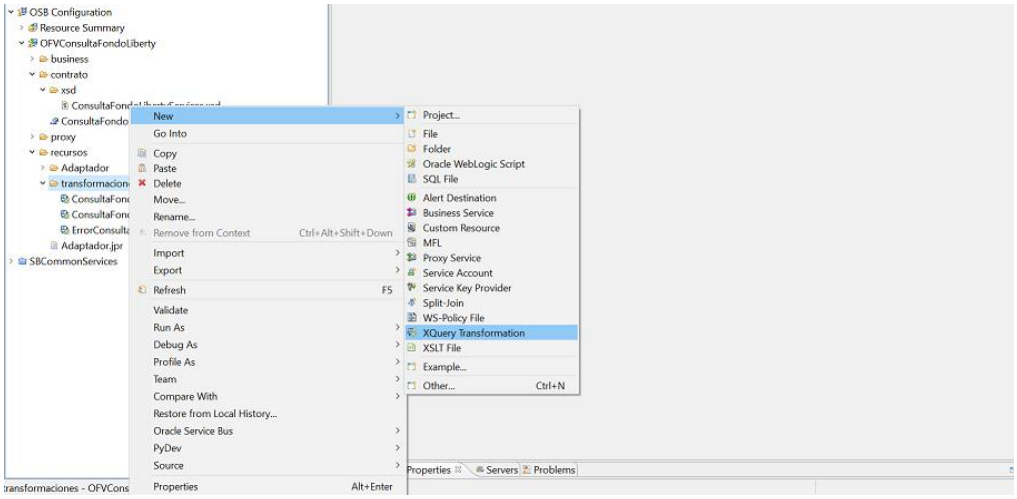

Figura 40. Creación de transformaciones (Fuente: Autor).

2. Se elige un nombre, y se oprime Next

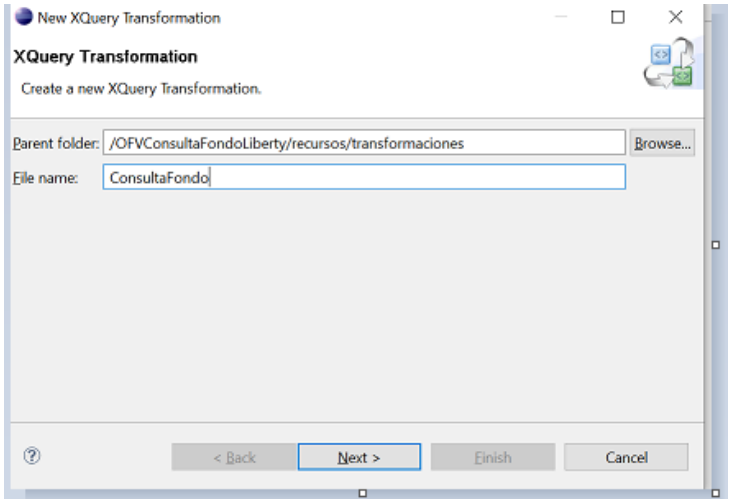

Figura 41. Elección de nombre para transformaciones (Fuente: Autor).

3. Seleccionar la opción XML, y buscar la XSD que se elaboró, posterior se elige la opción request o response dependiendo la transformación que se quiere realizar, es el mismo procedimiento para la xsd del adaptador, y se oprime Add y Next

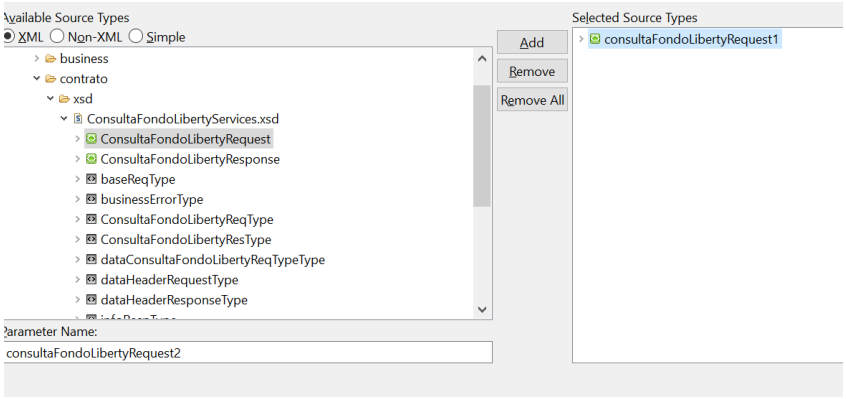

Figura 42. Búsqueda del elemento para agregar a la transformación (Fuente: Autor).

4. En la interfaz gráfica que nos muestra, se unen los campos del mismo nombre de las XSD, simultáneamente por cada conexión que se realiza en la pestaña source se auto genera código de este procedimiento.

- TRANSFORMACION DEL REQUEST

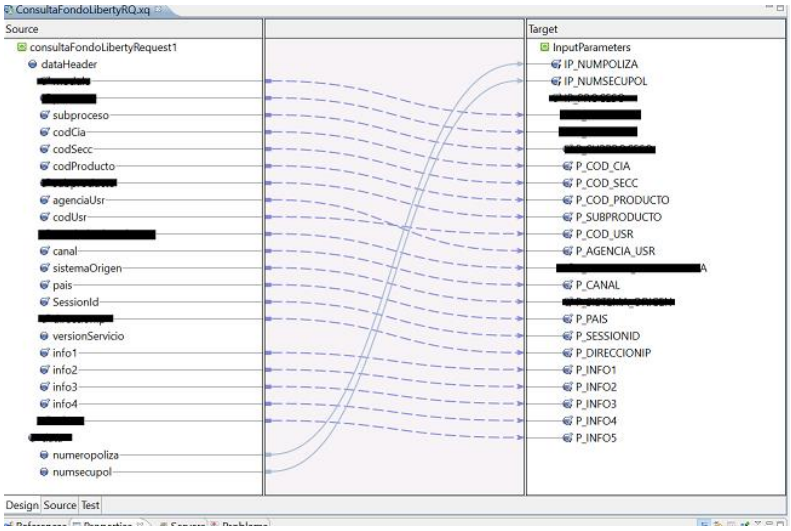

Figura 43. Grafica de mapeos del request (Fuente: Autor).

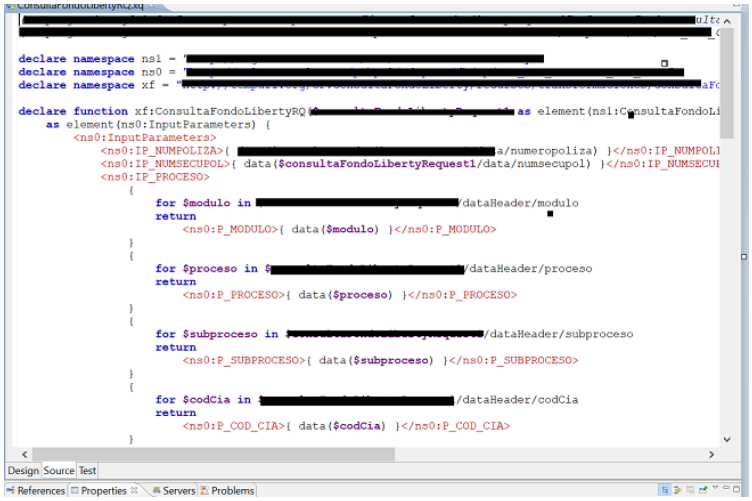

Figura 44. Código generado por el mapeo (Fuente: Autor).

- TRANSFORMACION DEL RESPONSE

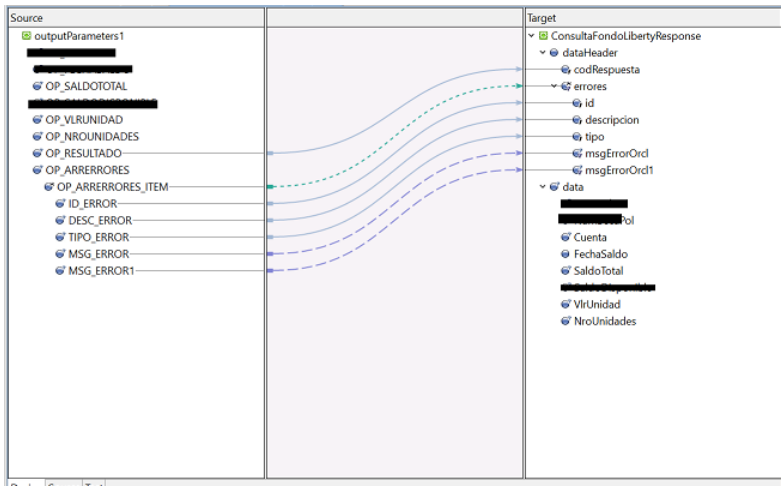

Figura 45. Gráfico de mapeo del response (Fuente: Autor).

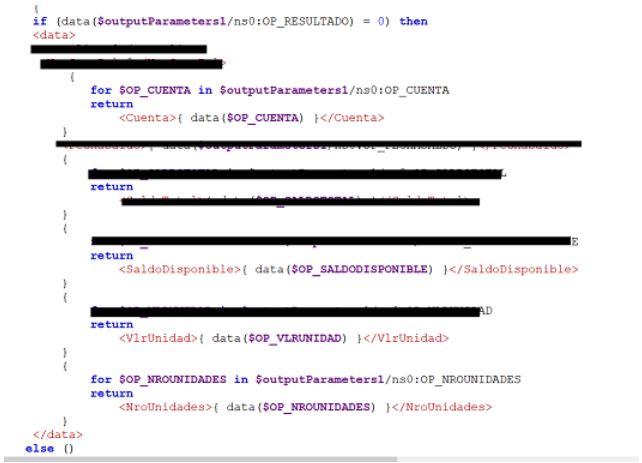

Figura 46. Código generado por el mapeo del response (Fuente: Autor).

Para el caso del response no se observa en la interfaz gráfica mapeado los valores de Data, pero en el código se observan que, si lo están, esto ocurre debido a que existe una condicional “if” que solo permite visualizar esos datos si el codRespuesta es cero.

- TRANSFORMACION DEL VECTOR DE ERRORES

  Para están transformación se crean las variables:

  A. descripcionError.

  B. codeError.

  C. MessageID

  D. msgError

  E. msgError1

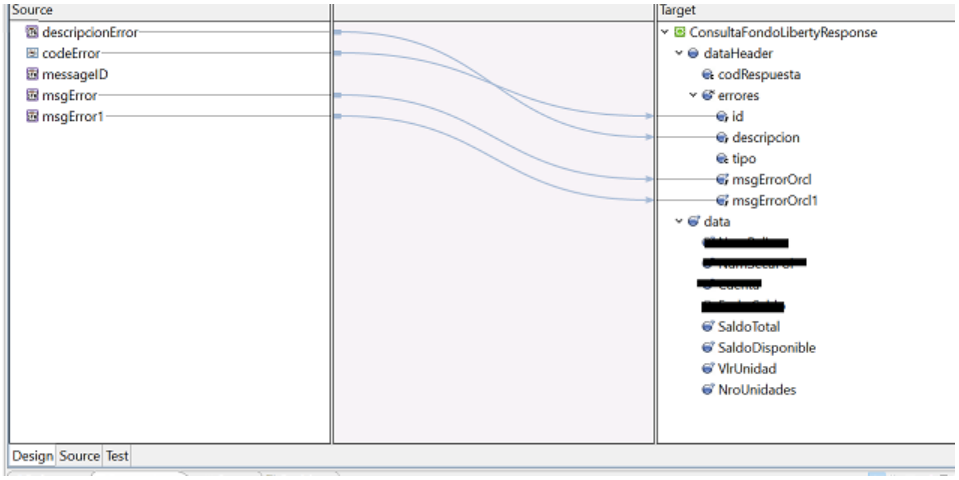

Figura 47. Grafico del mapeo del error (Fuente: Autor).

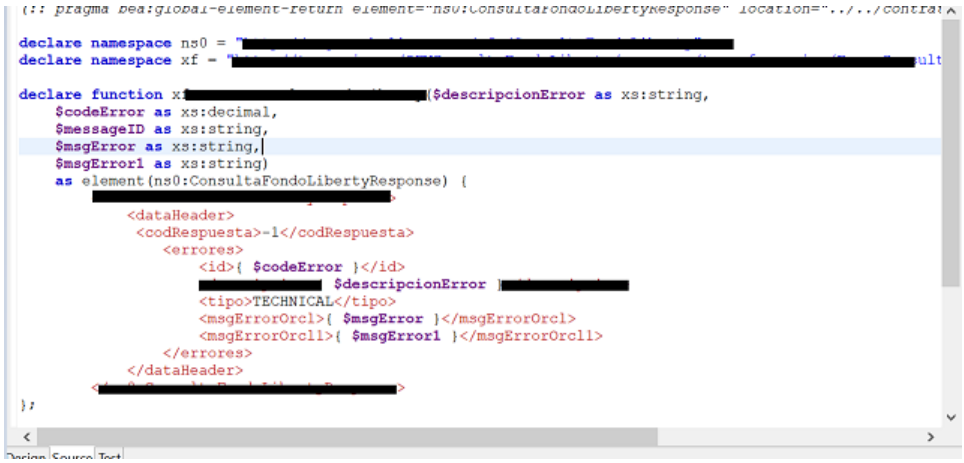

Figura 48. Código generado del mapeo del error (Fuente: Autor).

Con la creación de las variables anteriormente expuesto se busca que cuando exista un error, se conserve la estructura que esta estandarizada y para que los datos se puedan parametrizar en los campos de la XSD.

8.7. Creación del proxy service:

El proxy service es muy importante debido a que este contiene el flujo del servicio donde se debe definir la operación de la base de datos en el elemento routing, se declara una variable llamada Body, la cual toma el valor de la petición, esta después se valida con el cuerpo del request en el elemento Validete, si la variable Body no cumple las condiciones de estructura el flujo toma el camino del error handler, donde la variable Body toma el valor de error en el response e ingresa a la transformación del mismo, pero de ser correcto ingresa a la transformación del request y toma esa estructura. Despues Body toma el valor del response que viene de la base de datos, por consiguiente, se ingresa a la transformación del response para ser visualizada. Todos los mensajes tanto de response como de request están asociados a un servicio de auditoria, pero IBM no tiene acceso, y tampoco lo construyo por lo tanto no se tienen pruebas.

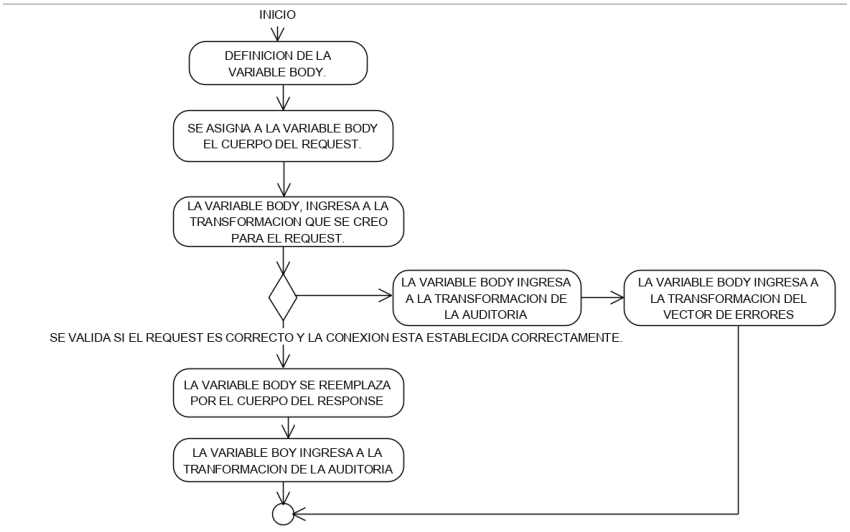

Figura 49. Flujo del proxy services (Fuente: Autor).

1. Sobre la carpeta Proxy, se da click derecho, New Proxy Service

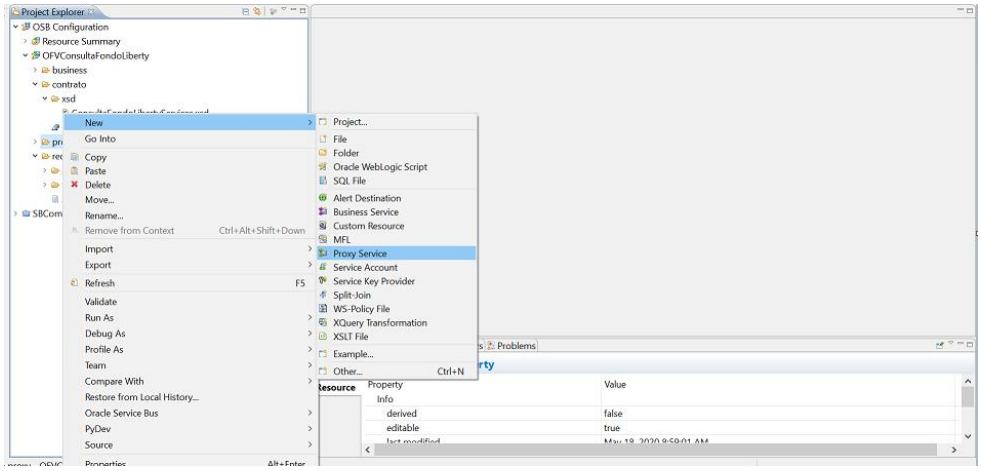

Figura 50. Creación del proxy services (Fuente: Autor).

2. Después de esto se asigna la WSDL que se creó.

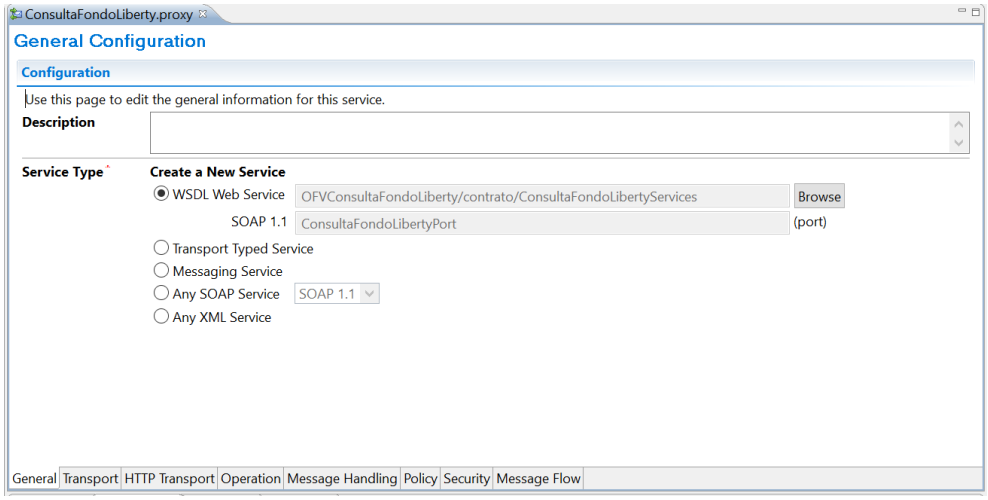

Figura 51. Grafica de la pestaña general del proxy services (Fuente: Autor).

3. En la pestaña de Message Flow se define el flujo del servicio

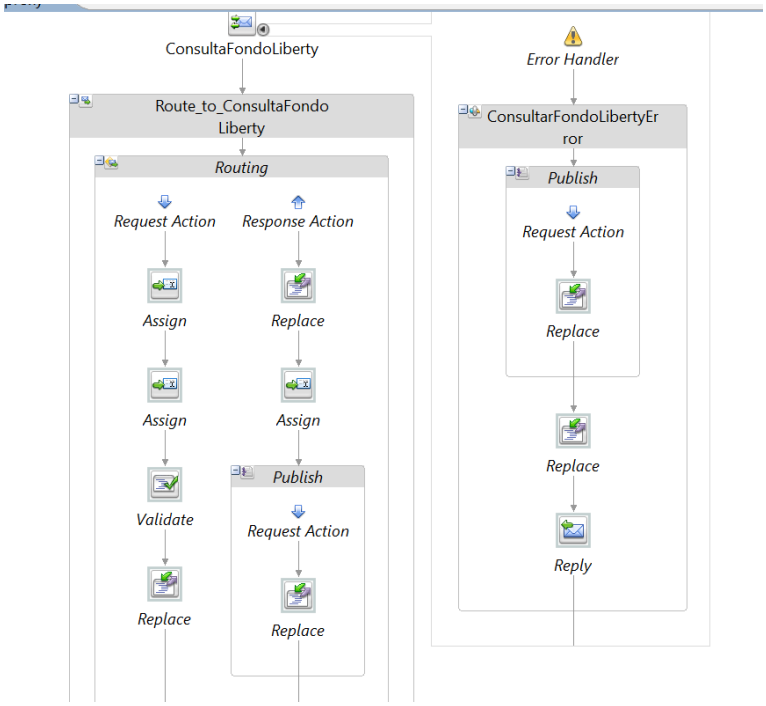

Figura 52. Grafica del message flow del proxy services (Fuente: Autor).

- **Routing**: En este recuadro se da todo el flujo del servicio debido a que se agrega el business service que se generó del adaptador y se elige el stored procedure que se eligio.

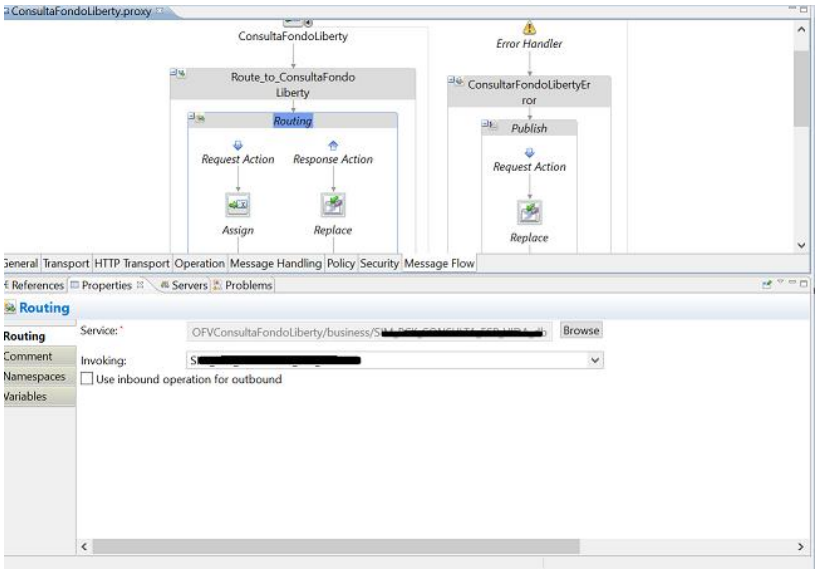

Figura 53. Grafica de configuración del routing (Fuente: Autor).

- **Assign**: En esta se asigna a la variable body el cuerpo del request y del response, dependiendo la rama en la que se encuentre.

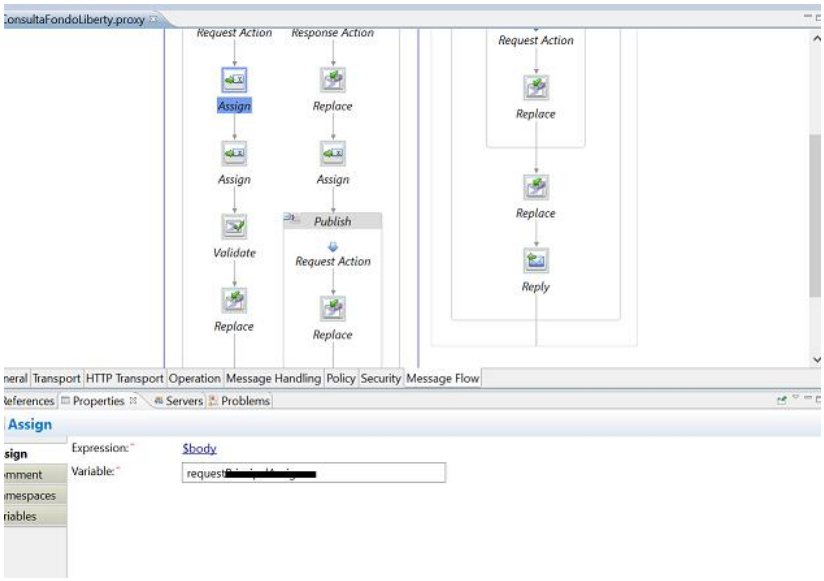

Figura 54. Grafica de configuración del Assign (Fuente: Autor).

- **Replace:** Este se asigna la variable Body a la transformación del request, response o error según corresponda.

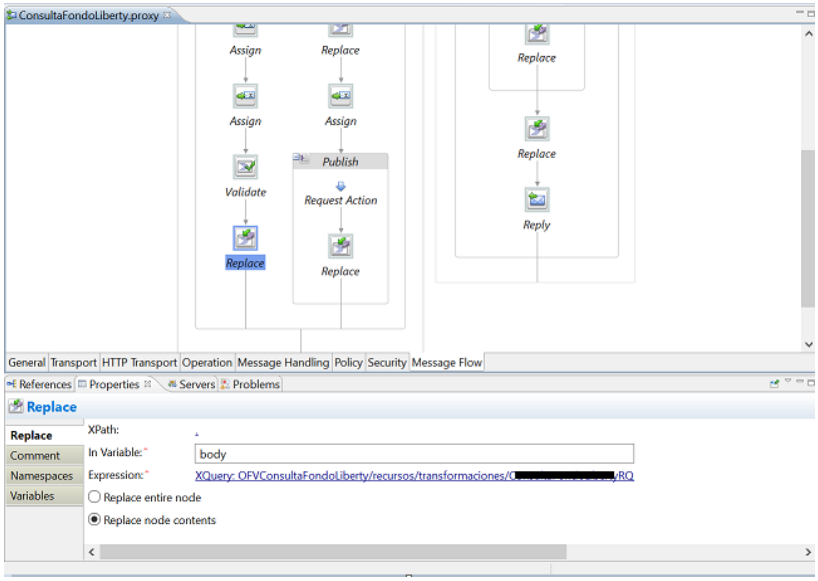

Figura 55. Grafica de configuración del Replace (Fuente: Autor).

- **Publish:** Este se le asigna la auditoria del servicio.

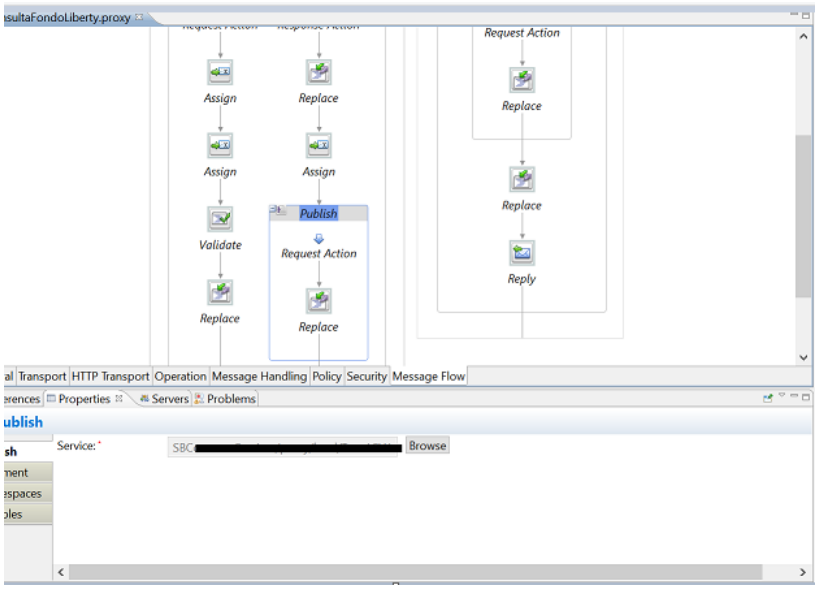

Figura 56. Grafica de configuración del Publish (Fuente: Autor).

- **Error handler:** En este se observa el flujo, cuando se presenta un error.

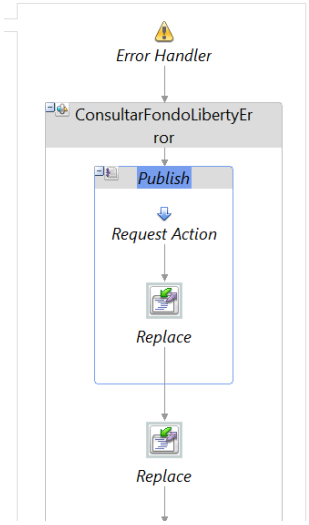

Figura 57. Grafica del flujo del Error (Fuente: Autor).

8.8. Generación del Jar y despliegue:

Realizar el despliegue es un paso importante en el desarrollo del servicio de integración debido a que permite realizar pruebas de la WSDL, este es un procedimiento sencillo en el cual el paso más importante es generar el archivo JAR, este contiene todas las carpetas y archivos que tiene el proyecto. Este proceso se realiza en eclipse, mientras que el despliegue se realiza en la consola de Oracle del Seguros Bolívar ingresando con credenciales autorizadas por el cliente. En la consola de Oracle se tiene que importar el

archivo JAR y siempre verificar que se encuentran todos los archivos del proyecto, además se debe tener extremo cuidado de no modificar otros archivos, esto puede generar graves problemas como modificaciones no autorizadas que afectan a otros proyectos.

1. Sobre la carpeta del proyecto, se presiona click derecho, export Oracle service bus-Configuration Jar

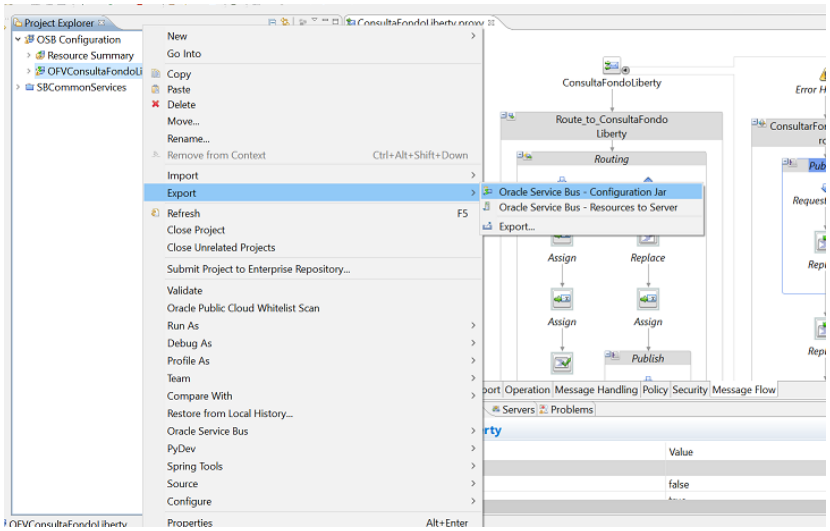

Figura 58. Generación del jar (Fuente: Autor).

2. Se asigna un nombre, y se presiona browse para indicar la dirección no se va a guardar el archivo.

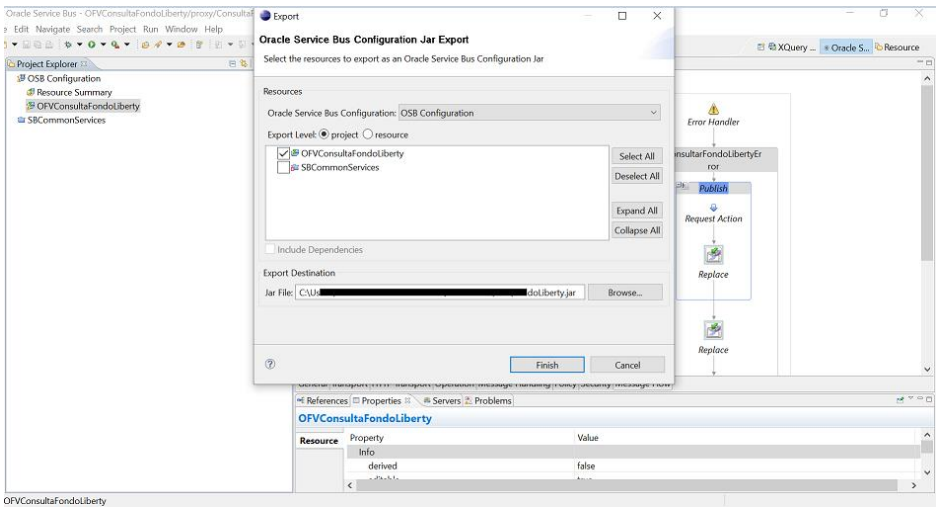

Figura 59. Dirección del jar (Fuente: Autor).

3. Se ingresa a la URL que está configurada para la consola del OSB, con el fin de ingresar usuario y contraseña teniendo en cuenta que se necesita estar conectado a la VPN de Seguros Bolívar.

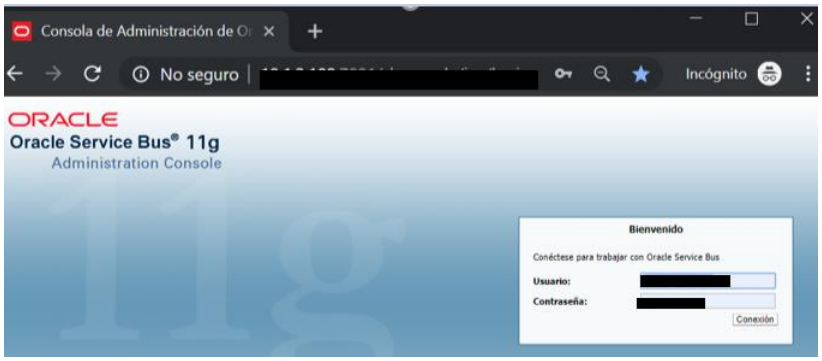

Figura 60. Ingreso a la consola del OSB (Fuente: Jonathan Sanchez, 2019).

4. Se Ubica el centro de Cambios y presiona el botón ***Crear***.

Figura 61. Ingreso al centro de cambios (Fuente: Jonathan Sanchez, 2019).

5. En la parte inferior presiona la opción ***Administrador del sistema.***

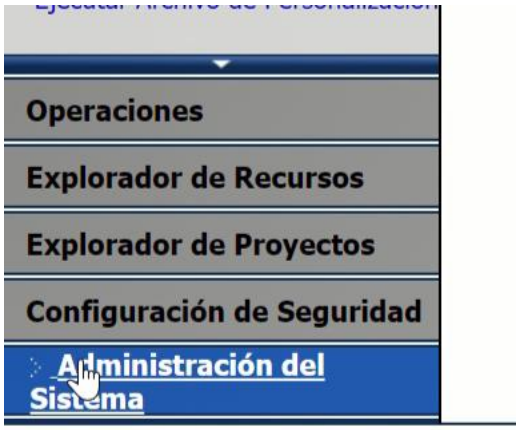

Figura 62. Ingreso a la administración del sistema (Fuente: Jonathan Sanchez, 2019).

6. Se Presiona botón ***"Seleccionar archivo"***, seleccione el archivo .jar que exporto del proyecto y presione el botón ***"Siguiente>>***

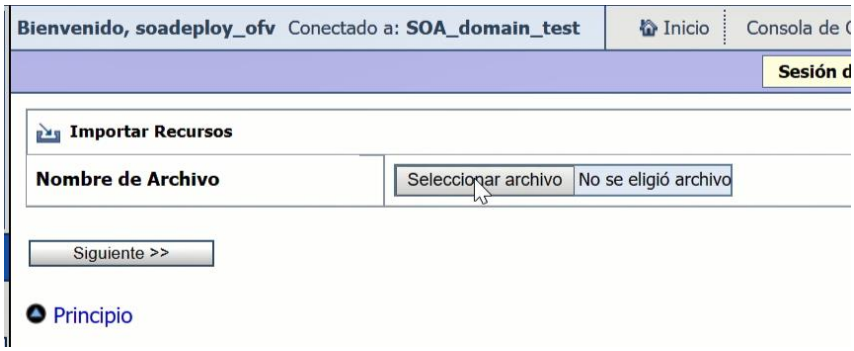

Figura 63. Búsqueda del JAR (Fuente: Jonathan Sanchez, 2019).

7. En la columna operaciones se puede observar que se creara un nuevo proyecto, presione el botón "***Importar"***.

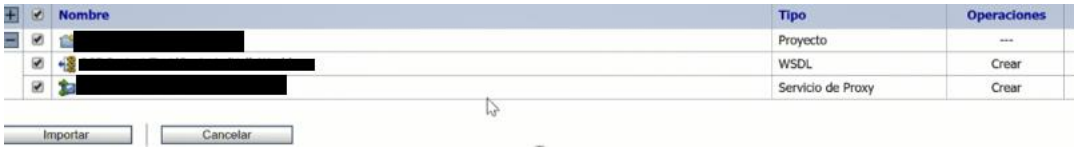

Figura 64. Elementos de JAR (Fuente: Jonathan Sanchez, 2019).

8. Presione el botón "***Activar****"*.

Figura 65. Grafica de activación de los cambios (Fuente: Jonathan Sanchez, 2019).

9. Se puede observar que el proyecto está activo.

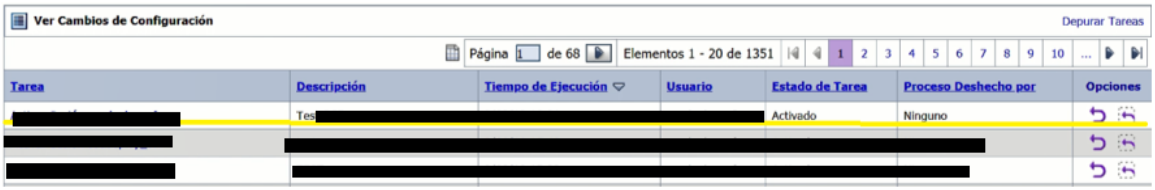

Figura 66. Validación de cambios (Fuente: Jonathan Sanchez, 2019).

8.9. Pruebas en SOAPUI-5.5.0

IBM para asegurar calidad en el software que desarrolla, plantea la necesidad de ejecutar pruebas para corregir posibles errores antes de entregar al cliente, para esto se usa el software de testing SOAPUI-5.5.0, debido a que está orientada aplicaciones que consuman servicios, y es de código abierto. Para las pruebas se va a tener en cuenta el campo codRespuesta debido a que, si tiene el valor de 0, debe mostrar la información que trae el servicio, pero si es -1 debe mostrar mensaje de error sea cual sea.

Para realizar pruebas del servicio de integración web se plantean cuatro casos de prueba

- Error de estructura: En este se envié un request que no cumple la estructura esperada.
- Resultado Exitoso: Para este caso se plantea la estructura la correcta del request, con datos que tienen información en la base de datos.
- Error de Negocio: Para este caso, se envía estructura correcta, pero con datos que no traen información de la base de datos.
- Error Técnico: Para este caso se desconecta la base de datos del servicio.

Las pruebas se realizan en el programa SOAPUI 5.5.0

1. Abre el programa y se oprime la opción SOAP, lo que despliega una ventana para buscar la WSDL, del proyecto

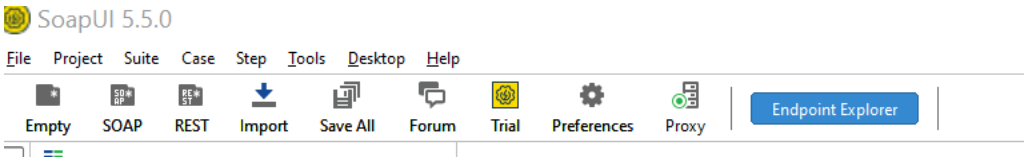

Figura 67. Menú de SOAPUI 5.5.0 (Fuente: Autor).

2. Se oprime Browse y para buscar la WSDL, del proyecto a importar

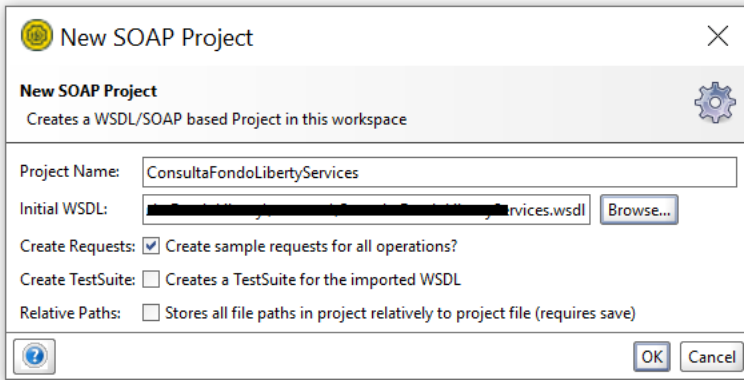

Figura 68. Ventana del WSDL importada (Fuente: Autor).

Figura 69. Request importado por WSDL (Fuente: Autor).

Ahora se visualiza los campos del request especificando cuales son opcionales y cuáles no.

3. Ahora se reemplazan los datos del request para generar el error de estructura, para este caso el número de póliza es el campo eliminado y se observa el resultado.

```
<soapenv:Body>
   <con:ConsultaFondoLibertyRequest>
      <dataHeader>

         <versionServicio>1.0</versionServicio>

      </dataHeader>
      <data>

         <numsecupol>0</numsecupol>
      </data>
   </con:ConsultaFondoLibertyRequest>
</soapenv:Body>
soapenv:Envelope>
```

Figura 70. Request del error de estructura (Fuente: Autor).

```
<con:ConsultaFondoLibertyResponse>
   <dataHeader>
      <codRespuesta>-1</codRespuesta>
      <errores>
         <id>2040</id>
         <descripcion>Error general al invocar el servicio</descripcion>
         <tipo>TECHNICAL</tipo>
         <msgErrorOrcl>7701879134673525864--7d8a0d8e.1727554ad46.-7e7f</msgErrorOrcl>
         <msgErrorOrcll>OSB Validate action failed validation</msgErrorOrcll>
      </errores>
   </dataHeader>
</con:ConsultaFondoLibertyResponse>
oapenv:Body>
```

Figura 71. Response de error de estructura (Fuente: Autor).

Como se observa el codRespuesta es -1, lo que indica que se encuentra en error y al observa el campo msgErrorOrcll indica que el OSB presenta falla de validación.

4. Resultado exitoso: para este caso se envía un request con la estructura y datos correctos.

```
<soapenv:Envelope [redacted]
   <soapenv:Header/>
   <soapenv:Body>
      <con:ConsultaFondoLibertyRequest>
         <dataHeader>

            <versionServicio>1.0</versionServicio>

         </dataHeader>
         <data>
            <numeropoliza>[redacted]/numeropoliza>
            <numsecupol>0</numsecupol>
         </data>
      </con:ConsultaFondoLibertyRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Figura 72. Request del caso exitoso (Fuente: Autor).

Figura 73. Response del caso exitoso (Fuente: Autor).

Como se observa el campo codRespuesta es 0 lo que indica que el resultado es correcto, además se observa información en los campos de data del request.

5. Error de negocio: para este caso se ingresan datos erróneos.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:con="http://seguro
   <soapenv:Header/>
   <soapenv:Body>
      <con:ConsultaFondoLibertyRequest>
         <dataHeader>

            <versionServicio>1.0</versionServicio>

         </dataHeader>
         <data>
            <numeropoliza>1</numeropoliza>
            <numsecupol></numsecupol>
         </data>
      </con:ConsultaFondoLibertyRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Figura 74. Request del error de negocio (Fuente: Autor).

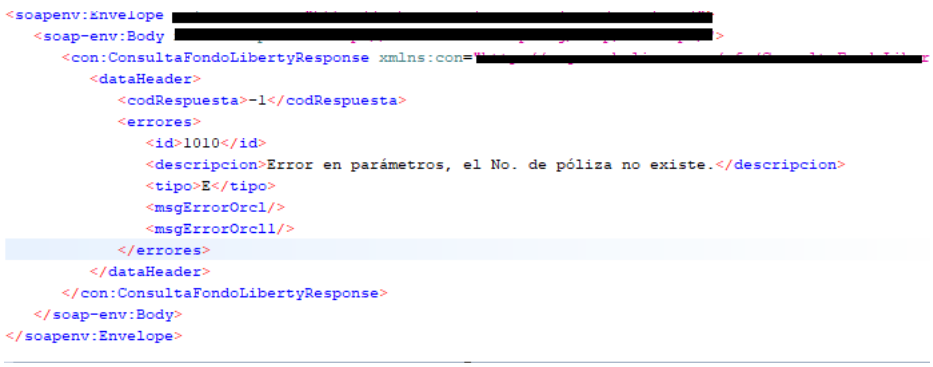

Figura 75. Response del error de negocio (Fuente: Autor).

Como se observa el codRespuesta es -1, además en la descripción del error se observa que indica que el número de póliza no existe.

6. Erro Técnico: para este caso se elimina el Business Service y se realiza el despliegue, con esto se logra que se desconecte la base de datos.

Figura 76. Request del error técnico (Fuente: Autor).

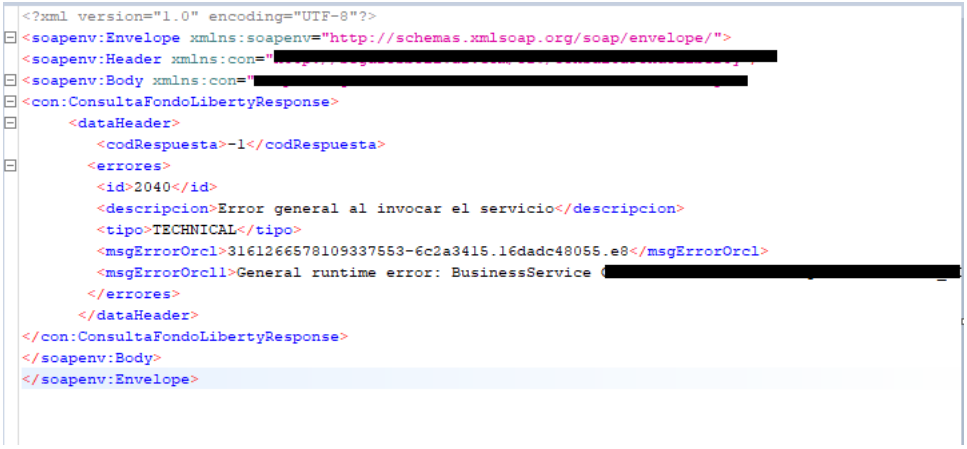

Figura 77. Response del error técnico (Fuente: Autor).

Como se observa el codRespuesta es -1, además el msgErrorOrcll indica que error en el business service lo que indica que la base de datos se desconectó.

Estos casos de prueba se plasman en un documento establecido por IBM, el cual establece el pan de pruebas y permite tener el control de versiones.

Nombre:	Plan de pruebas para el servicio OFVConsultaFondoLiberty	

Histórico de versiones:

Versión	Fecha	Autor	Comentarios
1.0	4/10/2019	Santiago Caceres Silva	Se realizan pruebas unitarias para el servicio OFVConsultaFondoLiberty

Figura 78. Página de inicio del formato de pruebas unitarias (Fuente: Autor).

Plan de Pruebas		Regresar al índice
	PRUEBAS UNITARIAS	
URL	https://am[redacted]	
WSDL	https://ambi[redacted]	
Prueba	Caso de prueba	Descripción
	consultaCobros	
CP03-PU01	Error de estructura	Se consume servicio generando una condición de error de estructura.
CP03-PU02	Resultado exitoso	Se consume servicio generando una condición de éxito.
CP03-PU03	Error de negocio	Se consume servicio generando una condición de error de negocio.
CP03-PU04	Error técnico	Se consume servicio generando una condición de error técnico.

Figura 79. Plan de pruebas del documento de pruebas unitarias (Fuente: Autor).

8.10. Documentación:

Para Seguros Bolívar es necesario generar un documento que, de especificación técnica, la cual indica cuales son los campos del servicio de request y response, la cardinalidad, el tipo de dato y el mapeo.

Aplicación Proveedora:		XML						Modo de Ejecución		Synchronous Req/Resp
Entrada	Protocolo	JDBC	Formato	XML		Salida	Protocolo	JDBC	Formato	XML
Namespaces		[redacted]								
Política a Implementar del Encabezado:		N/A				Características del Encabezado:		N/A		

Descripción de Campos de Entrada:

Orden	Nombre de Campo	Cardinalidad	Tipo	Long. Max.	Formato	Dominio de Valores	Justificación o Alineación	Carácter de Relleno	Valor por Defecto	Es Auditable	Descripci
1.1	inputParameters	0..n	contenedor	N/A	N/A	N/A	N/A	N/A	N/A	SI	Contenedor
1.1.1	IP_PROCESO	0..n	contenedor	N/A	N/A	N/A	N/A	N/A	N/A	SI	Contenedor
1.1.1	P_MODULO	0..1	string	N/A	N/A	N/A	N/A	N/A	N/A	SI	modulo
1.1.2	P_PROCESO	0..1	decimal	N/A	N/A	N/A	N/A	N/A	N/A	SI	proceso
1.1.3	P_SUBPROCESO	0..1	decimal	N/A	N/A	N/A	N/A	N/A	N/A	SI	subProceso
1.1.4	P_COD_CIA	0..1	decimal	N/A	N/A	N/A	N/A	N/A	N/A	SI	codCia
1.1.5	P_COD_SECC	0..1	decimal	N/A	N/A	N/A	N/A	N/A	N/A	SI	codSecc
1.1.6	P_COD_PRODUCTO	0..1	decimal	N/A	N/A	N/A	N/A	N/A	N/A	SI	codProducto
1.1.7	P_SUB_PRODUCTO	0..1	decimal	N/A	N/A	N/A	N/A	N/A	N/A	SI	subProducto
1.1.8	P_COD_URS	0..1	string	N/A	N/A	N/A	N/A	N/A	N/A	SI	codUrs
1.1.9	P_ENTIDAD_COLOCADERA	0..1	decimal	N/A	N/A	N/A	N/A	N/A	N/A	SI	entidadColocadora
1.1.10	P_CANAL	0..1	decimal	N/A	N/A	N/A	N/A	N/A	N/A	SI	canal
1.1.11	P_SISTEMA_ORIGEN	0..1	decimal	N/A	N/A	N/A	N/A	N/A	N/A	SI	sistemaOrigen
1.1.12	P_PAIS	0..1	decimal	N/A	N/A	N/A	N/A	N/A	N/A	SI	pais
1.1.13	P_DIRECCIONIP	0..1	string	N/A	N/A	N/A	N/A	N/A	N/A	SI	direccionIp

Figura 80. Documento de especificación técnica (Fuente: Autor).

8.11. Solicitud de peer review:

En los estándares de IBM, siempre se solicita revisión de par, de cualquier desarrollo que se realice, para esto se usa la herramienta jira y se crea una tarea que solicite el peer review a una persona del equipo de trabajo.

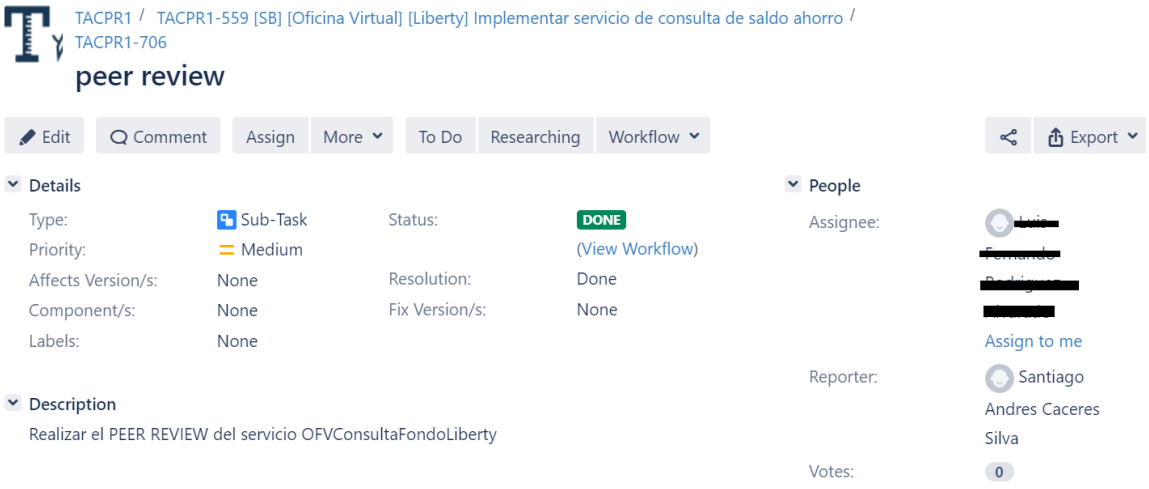

Figura 81. Solicitud de peer review en Jira.

Para esta revisión el par revisor genera un documento con los errores encontrados

Registro de Issues

ID	Servicio Involucrado	Componente	Descripción del Issue	Impacto	Reproceso	Fecha Origen	Registrado Por	Asignado
1	OFVConsultaFondoLiberty	Entrega	El directorio de entrega no tiene la estructura adecuada. \SB\OSB\Entregables\Test\OFV\OFVConsultaFondoLiberty\entregables02octubre2019	Bajo	Bajo	08/10/2019	Luis Rodriguez	
2	OFVConsultaFondoLiberty	SBSwF - Reporte Pruebas OFVConsultaFondoLiberty-v1.0.xls	Los enlaces de la pestaña "PlanPruebas" no direccionan correctamente a las pestañas de cada una de las pruebas.	Bajo	Bajo	08/10/2019	Luis Rodriguez	
3	OFVConsultaFondoLiberty	SBSwF - Reporte Pruebas OFVConsultaFondoLiberty-v1.0.xls	La URL en la pruebas unitarias no corresponden con las del servicio	Bajo	Bajo	08/10/2019	Luis Rodriguez	

Figura 82. Errores encontrados por la persona que realiza el peer review (Fuente: Autor).

Como se observa se especifica el servicio involucrado, el componente, la descripción del error, para este caso se cometieron errores en documentación y por lo tanto el indicie de impacto es bajo.

8.12. Entrega:

Para terminar el desarrollo se debe realizar entrega al equipo de QA (Quality Assurance), para que ellos lo hagan llegar a la persona encargada del cliente, para esto se crea una tarea en Jira especificando la ruta en repositorio GIT donde queda el entregable que en este caso es el archivo JAR, generado para el despliegue y la documentación.

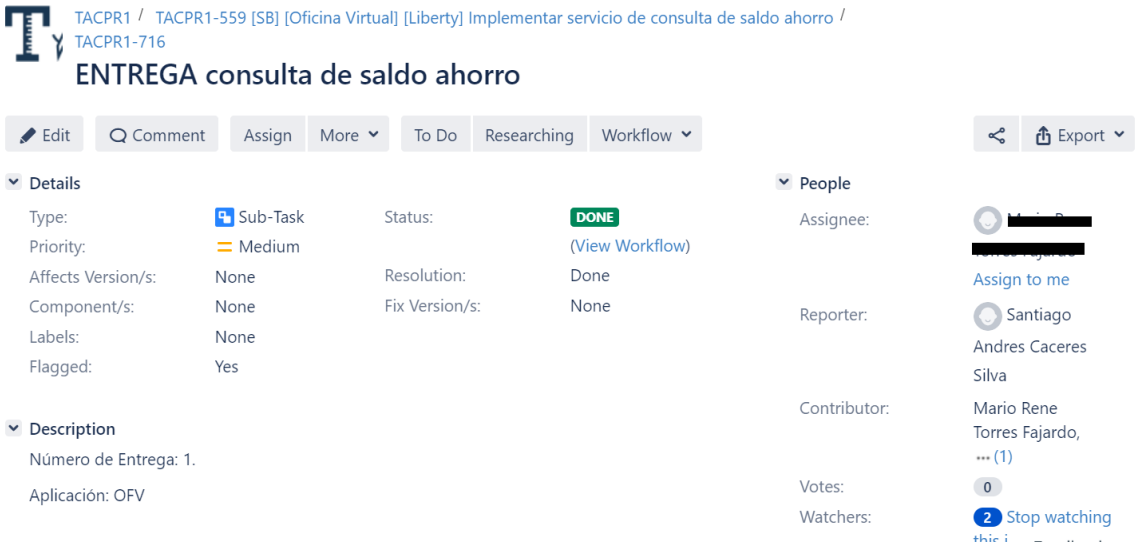

Figura 83. Solicitud de entrega al equipo de QA (Fuente: Autor).

Componentes:

Documentación:

-

Ruta de documentación:

OSB\Documentacion\OFV\OFVConsultaFondoLiberty\SBSwT-EspecficacionOFVConsultaFondoLiberty.xlsx

Ruta de Entregables:

OSB\Entregables\Test\OFV\OFVConsultaFondoLiberty

Ruta de Fuentes:

OSB\Fuentes\OFV\OFVConsultaFondoLiberty

-

Rama Git: feature/Dev

Figura 84. Especificación de la entrega (Fuente: Autor).

Con esto se da por terminado el desarrollo de parte del equipo de integración.

Capítulo 9
Conclusiones

- El uso de bus de integración permite darle un valor agregado al negocio, debido a que el desarrollo es más rápido lo que conlleva a que la aplicación este en producción en menor tiempo, también porque ahorra consumo de infraestructura TIC y costos de mantenimiento.
- En la etapa de pruebas se genera un error por cada circunstancia presentada, lo que permite fácil solución si se llega a presentar un error, debido a que se tiene el conocimiento que es a nivel de bus de integración y no a nivel de backend, por consiguiente, el tiempo de solución del incidente es menor.
- IBM es una empresa que implementa nuevas metodologías de trabajo pensando en el cliente, por lo tanto, para este proyecto se utiliza agile scrum, la cual es una prenda de garantía al cliente debido a que al final de cada sprint se puede observar avances funcionales del proyecto, porque todos los equipos trabajan simultáneamente. Por otro lado, esta metodología permite tener mayor control del desarrollado debido a sus tres preguntas ¿Qué hice ayer? ¿Qué voy a hacer hoy? ¿Qué bloqueantes tengo?, lo que permite generar una solución en el menor tiempo posible, y el PM pueda estimar si hay retrasos si el bloqueante es crítico.
- La herramienta Jira es de gran ayuda para tener control de actividades de proyecto debido a que permite observar los estados de la actividad, lo que con lleva a que si se presenta un problema se solucione rápido, y también para saber el tiempo trabajado en la tarea, información útil para los Project Management.
- En el proyecto expuesto se observa como el OSB, permite el uso de diferentes protocolos de comunicación, por ejemplo, del backend al bus de servicios se usa el protocolo HTTP, y del bus de integración a la base de datos se usa el protocolo JDBC, con una mayor eficiencia y sin preocupación que alguna parte de la comunicación no soporte el protocolo.
- El bus de integración de servicios web, presenta un entorno de desarrollo en su mayoría gráfico y de menor codificación, esto hace que el desarrollador tenga un entorno más amigable y con menor posibilidad de error, lo que permite que el proyecto avance de manera rápida.
- El manejo de GIT es bastante útil debido a que evita conflictos de código con su trabajo en ramas, esto previene riesgos porque si se trabajara sobre la misma y dos desarrolladores trabajen el mismo proyecto uno puede borrar cosas del otro y se tendría un algoritmo poco funcional.
- Realizar la práctica profesional en una empresa como IBM, permite un gran crecimiento profesional y personal, en razón a que se trabaja con bastantes

tecnologías para el desarrollo web basándose en el cumplimiento de estándares de calidad y satisfacción para el cliente dando un valor a la tecnología, adicionalmente permite realizar cursos y obtener certificaciones, en distintos temas como, por ejemplo: Big Data, Machinne Learning, Watson, Design thinking, metodologías agiles, Data sciencie entre otras; por otro lado trabajar buscando objetivos genera un grado responsabilidad y compromiso mayor.

Lista de referencias

[1] Kress, J., Maier, B., & Normann, H. (2013, julio). Enterprise Service Bus. Recuperado de https://www.oracle.com/technical-resources/articles/middleware/soa-ind-soa-esb.html

[2] Enterprise Service Bus(E.S.B). (2019, 18 julio). Recuperado de https://www.ibm.com/cloud/learn/esb#toc-esb-and-so-vSddjRBt

[3] Safelayer Marketing D, S. (2016, 26 abril). Enterprise Service Bus (ESB): la infraestructura de interconexión para SOA. Recuperado 9 enero, 2020, de https://www.safelayer.com/es/recursos/69-articulos/integracion-de-servicios-de-seguridad/476-enterprise-service-bus-esb-la-infraestructura-de-interconexion-para-soa

[4]Mario Guerrero, M. (2018, 30 julio). 5 cosas que debes saber sobre la Arquitectura Orientada a Servicios. Recuperado 9 enero, 2020, de https://www.megapractical.com/blog-de-arquitectura-soa-y-desarrollo-de-software/por-que-tus-servicios-de-atencion-al-cliente-necesitan-integracion-0

[5] Arquitectura Orientada a Servicios (SOA). (2018, 12 julio). Recuperado 9 enero, 2020, de https://www.accenture.com/t20150527T210642__w__/es-es/_acnmedia/Accenture/Conversion-Assets/DotCom/Documents/Local/es-es/PDF_2/Accenture-SOA.pdf

[6] Metodología SCRUM para desarrollo de software a medida. (s.f.). Recuperado 9 enero, 2020, de https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html

[7]ORACLE. (2011). Oracle Service Bus. Recuperado 9 enero, 2020, de https://www.avanttic.com/pdf/Oracle_Service_Bus_Enrique_Martin.pdf

[8] Jonathan S. S. (2019). OSB11g - Guia de instalación y despliegue.

[9] Vicepresidencia de Tecnología Helm Bank(2016) Lineamientos de desarrollo OSB 12 C_V1.2

[10] Daniel V. U.(2019). IBMSwF - Guia del desarrollador ITAU.

[11]Gustavo B.(2013)¿Qué es GitHub y para que se utiliza? https://www.hostinger.co/tutoriales/que-es-github/

[12] SQL Procedimientos Almacenados de SQL Server, https://www.w3schools.com/sql/sql_stored_procedures.asp

[13]Oracle conceptos y arquitectura de fusión middleware para Oracle service bus https://docs.oracle.com/cd/E17904_01/doc.1111/e15020/architecture_overview.htm#OSBCA140

[14] Oracle Guía del administrador de Fusion Middleware para Oracle Service Bus https://docs.oracle.com/cd/E23943_01/admin.1111/e15867/intro_console.htm#OSBAG799

[15]¿Para qué sirve Jira? https://www.atlassian.com/es/software/jira/guides/use-cases/what-is-jira-used-for#Jira-for-requirements-&-test-case-management
[16]Ruteo de servicios Web con Oracle Service Bus (Parte 1), Deiby Gomez Robles https://www.oracle.com/technetwork/es/articles/soa/ruteo-de-servicios-web-part1 1991367-esa.html
[17] Oracle Service Bus, Enrique Martín Casado, Presales Manager, https://www.avanttic.com/pdf/Oracle_Service_Bus_Enrique_Martin.pdf