



PROPUESTA PARA LA INTEROPERABILIDAD DE PROCESOS DE NEGOCIO EN SISTEMAS DE INFORMACIÓN HOSPITALARIOS BAJO UNA ARQUITECTURA ORIENTADA A SERVICIOS



ANEXOS

**Peter Emerson Pinchao Solís
Nicolás Rubén Caballero Ortega**

Universidad del Cauca

**Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Grupo de Ingeniería Telemática-GIT
Línea de Investigación en Aplicaciones y Servicios Sobre Internet
Popayán, Marzo de 2011**



Tabla de Contenidos

| | |
|---|-----------|
| LISTA DE FIGURAS | 3 |
| ÍNDICE DE TABLAS | 7 |
| A DISEÑO DE LA ARQUITECTURA BAJO EL MODELO DE REFERENCIA SOA. | 8 |
| A.1. ANÁLISIS ORIENTADO A SERVICIOS. | 8 |
| A.1.1. Definir Requerimientos de Negocio..... | 8 |
| A.1.2. Identificar Sistemas Automatizados..... | 9 |
| A.1.3. Modelado de servicios candidatos..... | 10 |
| A.2. DISEÑO ORIENTADO A SERVICIOS | 19 |
| A.2.1. Composición de SOA. | 20 |
| A.2.2. Diseñar los Servicios de Negocio..... | 22 |
| A.2.3. Diseñar los servicios de aplicación..... | 25 |
| A.2.4. Diseño de los servicios de negocio centrados en tareas..... | 27 |
| A.2.5. Diseño de Procesos de Negocio Orientados a Servicios..... | 30 |
| A.3. DESARROLLO DE SERVICIOS..... | 34 |
| A.4. PRUEBAS DE SERVICIOS..... | 35 |
| A.5. DESPLIEGUE DE SERVICIOS..... | 35 |
| A.6. ADMINISTRACIÓN DE SERVICIOS..... | 36 |
| B COMPARACIÓN DE ESB DE LIBRE DISTRIBUCIÓN (ANEXO B). | 37 |
| B.1. OPENESB..... | 37 |
| B.2. CHAINBUILDER..... | 37 |
| B.3. FUSE ESB..... | 38 |
| B.4. PETALS ESB..... | 39 |
| B.5. JBOSS ESB..... | 39 |
| B.6. APACHE SYNAPSE..... | 40 |
| B.7. APACHE SERVICE MIX..... | 41 |
| B.8. WSO2 ESB..... | 41 |
| B.9. MULE ESB..... | 42 |
| B.10. JEE ESB..... | 43 |
| B.11. CELTIX..... | 43 |
| B.12. RESUMEN DE RESULTADOS..... | 44 |
| C INSTALACIÓN DE OPEN ESB | 45 |
| C.1. DESCARGAR OPENESB 2.2 | 45 |
| C.2. INSTALACIÓN DE OPENESB..... | 45 |
| C.3. INTRODUCCIÓN A GLASSFISH ESB 2.2 | 50 |
| C.3.1. IDE Netbeans con capacidades SOA..... | 50 |
| C.3.2. Servidor GlassFish con funcionalidad ESB..... | 52 |
| C.4. CONFIGURACIÓN DE POOL DE CONEXIONES POSTGRESQL..... | 58 |
| D INSTALACIÓN DE HL7 BUILDING COMPONENT (BC) | 66 |
| D.1. PRERREQUISITOS | 66 |



| | |
|--|------------|
| D.1.1. Instalación de HL7 Encoder | 66 |
| D.1.2. Instalación de Plugins para Netbeans IDE..... | 68 |
| D.2. INSTALACIÓN DE HL7 BUILDING COMPONENTS | 72 |
| D.3. CREACIÓN Y VALIDACIÓN DE UN WSDL EMPLEANDO HL7 ENCODER PARA CREACIÓN DE UN MENSAJE HL7. | 73 |
| D.4. Prueba del codificador desde Component Aplicación..... | 85 |
| D. 4 CODIFICADOR GFORSE PARA MAPEAR HL7 V2 A HL7 V3 | 89 |
| D.5 CONSTRUCCIÓN DE UN SERVICIO WEB DESDE UN EJB PARA HABILITAR LA TRASFORMACIÓN..... | 96 |
| E PROTOCOLO HL7 EN LA ARQUITECTURA ORIENTADA A SERVICIOS..... | 98 |
| F ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LOS APLICATIVOS AUDITORIA Y TELE CONSULTA..... | 100 |
| F.1. ANÁLISIS. | 100 |
| F.1.1. Descripción del problema. | 100 |
| F.1.2. Diagrama de casos de Uso..... | 100 |
| F.1.3 Modelo Conceptual. | 106 |
| F.1.4. Diagrama de Secuencias..... | 106 |
| F.2.1. Diagrama de paquetes. | 108 |
| F.2.2. Modelo físico..... | 109 |
| F.3. IMPLEMENTACIÓN. | 110 |
| F.3.1. Módulo EJB. | 113 |
| F.3.2. Persistencia con JPA. | 114 |
| F.3.3. Lógica del Negocio en el Módulo EJB Servicio de Aplicación: Administrar Observaciones..... | 123 |
| F.3.4. Servicios web mediante EJB. | 126 |
| G PRUEBAS DE RENDIMIENTO Y ESCALABILIDAD. | 129 |
| G.1. CONFIGURACIÓN DE CLÚSTER OPENSEB Y PRUEBAS DE ESCALABILIDAD. | 129 |
| G.1.1. Configuración de GlassFish Soporte de Clúster..... | 129 |
| G.1.2. Escenarios de Evaluación de la Escalabilidad..... | 132 |
| H GUÍA PARA LA IMPLEMENTACIÓN DE CANAL SOAP EN MITH..... | 137 |
| H.1. DESCRIPCIÓN DE MIRTH. | 137 |
| H.2. ARRANQUE DEL SERVIDOR (MIRTH SERVER) Y DEL ADMINISTRADOR (MIRTH ADMINISTRATOR)..... | 137 |
| H.3 CREACIÓN DEL CANAL Y CONFIGURACIÓN DEL CONECTOR DE ENTRADA (SOURCE CONNECTOR)..... | 140 |
| H.4 CONFIGURACIÓN DEL TRANSFORMADOR (TRANSFORMER) | 143 |
| H.5 CREACIÓN Y CONFIGURACIÓN DEL EXTREMO DESTINO DEL CANAL (CHANNEL DESTINATION). | 150 |
| H.6 PUBLICACIÓN Y DESPLIEGUE DEL CANAL | 151 |
| H.6 PUBLICACIÓN Y DESPLIEGUE DE UN CANAL SOAP | 152 |
| I GUÍA DE IMPLEMENTACIÓN DE UN SERVICIO WEB CON TRASFORMACIÓN HL7 EMPLEANDO CHAINBUILDER (GETTING STARTED WITH CHAINBUILDER CONNECT)..... | 155 |
| REFERENCIAS..... | 157 |



LISTA DE FIGURAS

| | |
|--|----|
| FIGURA A.1. PROCESO AUDITORIA | 11 |
| FIGURA A.2. PROCESO TELE-CONSULTA. | 12 |
| FIGURA A.3. SERVICIOS CANDIDATOS. | 15 |
| FIGURA A.4. REPRESENTACIÓN DE LA COMPOSICIÓN DE PROCESO AUDITORIA. | 16 |
| FIGURA A.5. REPRESENTACIÓN DE LA COMPOSICIÓN DE PROCESO TELE-CONSULTA..... | 16 |
| FIGURA A.6 NIVELES DE SERVICIOS..... | 21 |
| FIGURA A.7. VISTAN DE COMPISICION DEL NUCLEO DE ESTANDARES PARA LA SOA..... | 21 |
| FIGURA A.8 VISTA DE LOS ESTENCIONES PARA LA DESCRIPCION DE SERVICIOS. | 22 |
| FIGURA A.9. SERVICIOS DE NEGOCIO Y SUS OPERACIONES..... | 24 |
| FIGURA A.10. JERARQUIA DE COMPOSICION DEL SERVICIO DE NEGOCIO DE AUDITORIARIA. | 25 |
| FIGURA A.11. DIAGRAMA DE SECUENCIA SERVICIO TELE CONSULTA. FLUJO COMUN. | 28 |
| FIGURA A.12. DIAGRAMA DE SECUENCA SERVICIO DE TELECONSULTA. PARAMETRIZADO..... | 29 |
| FIGURA A.13. DIAGRAMA DE SECUENCIA DE AUDITORIA..... | 29 |
| FIGURA A.14. SOLICITUS Y RESPUESTAS DE LOS SERVICIOS DE NEGOCIO. | 30 |
| FIGURA A.15. PROCESO AUDITORIA | 34 |
| FIGURA A.16. PROCESO TELE-CONSULTA | 34 |
| FIGURA C.1 ASISTENTE DE INSTALACIÓN OPENSEB..... | 45 |
| FIGURA C.2 PERSONALIZAR INSTALACIÓN | 46 |
| FIGURA C.3 TERMINOS DE LICENCIAMIENTO | 46 |
| FIGURA C.4 CONFIGURACIÓN DE ENTORNO JAVA..... | 47 |
| FIGURA C.5. CONFIGURACIÓN DE SERVICIOS..... | 48 |
| FIGURA C.6. RESUMEN DE CONFIGURACIÓN..... | 49 |
| FIGURA C.7. INSTALACIÓN DE OPENSEB..... | 49 |
| FIGURA C.8. INSTALACIÓN COMPLETA OPENSEB..... | 50 |
| FIGURA C.9. NETBEANS IDE..... | 51 |
| FIGURA C.10. CAPASIDADES PARA SOA NETBEANS IDE..... | 52 |
| FIGURA C.11. OPENSEB BASADO EN GLASSFISH..... | 53 |
| FIGURA C.12. OPENSEB JBI..... | 53 |
| FIGURA C.13. MOTOR BPEL EN OPENSEB..... | 54 |
| FIGURA C.14. INTERFACE DE CONFIGURACIÓN OPENSEB..... | 55 |
| FIGURA C.15. INTERFACE DE ADMINISTRACIÓN OPENSEB COMPONENTES JBI..... | 56 |
| FIGURA C.16. INTERFACE DE ADMINITRACION OPENSEB PROPIEDADES DE UN COMPONENTE JBI | 57 |
| FIGURA C.17. INTERFACE DE ADMINISTRACIÓN OPENSEB POOL DE CONEXIONES..... | 58 |
| FIGURA C.18. BASE DATOS POSTGRESQL | 59 |
| FIGURA C.19. DRIVES JDBC POSTRESQL..... | 60 |
| FIGURA C.20. AGRGAR DRIVER JDBC POSTGRESQL..... | 60 |
| FIGURA C.21. CONECCIÓN JDBC POSTGRESQL..... | 61 |
| FIGURA C.22. CREAR UN POOL DE CONEXIONES..... | 61 |
| FIGURA C.23. SELECCIONAR CONEXION PARA CREAR EL POOL DE CONEXIONES..... | 62 |
| FIGURA C.24. CONFIGURACIÓN DEL POOL DE CONEXIONES..... | 62 |
| FIGURA C.25. ARCHIVO DE CONFIGURACIÓN DE POOL..... | 63 |
| FIGURA C.26. CONFIGURAR RECURSO EN DE POOL EN GLASFISH..... | 64 |
| FIGURA C.27. POOL DE CONEXIONES POSTGRESQL..... | 65 |



| | |
|--|-----|
| FIGURA D.1. DESINSTALACION DEL CODIFICADOR. | 67 |
| FIGURA D.2. INSTALACIÓN DEL CODIFICADOR HL7..... | 67 |
| FIGURA D.3. SELECCIÓN DE LAS LIBRERIAS QUE COMPONEN EL CODIFICADOR. | 68 |
| FIGURA D.4. PROPIEDADES DEL CODIFICADOR HL7. | 68 |
| FIGURA D.5. NETBEANS IDE - ADMINISTRACIÓN DE COMPLEMENTOS..... | 69 |
| FIGURA D.6. NETBEANS IDE - AGREGAR PLUGINS | 69 |
| FIGURA D.7. NETBEANS IDE - ASISTENTE DE CONFIGURACIÓN DE PLUGINS. | 70 |
| FIGURA D.8. NETBEANS IDE - MENSAJE DE VALIDACIÓN DE FIRMAS DIGITALES. | 70 |
| FIGURA D.9. NETBEANS IDE - INTALACIÓN COMPLETA DE PLUGINS..... | 71 |
| FIGURA D.10. NETBEANS IDE - PLUGINS INSTALADOS. | 71 |
| FIGURA D.11. INSTALACIÓN DE HL7 BC | 72 |
| FIGURA D.12. AGREGAR COMPONENTE HL7..... | 72 |
| FIGURA D.13. PROPIEDADES DEL COMPONENTE HL7. | 73 |
| FIGURA D.14. COMPONENTE HL7 AGREGADO. | 73 |
| FIGURA D.15. NUEVO PROYECTO HL7 BUILDING COMPONENT..... | 74 |
| FIGURA D.16. NOMBRE DEL MODULO HL7 BC. | 74 |
| FIGURA D.17. NUEVO ESQUEMA XML..... | 75 |
| FIGURA D.18. SELECCION DE LA ESTRUCTURA DEL MENSAJE HL7. | 75 |
| FIGURA D.19. NOMBRE DE HL7BC_SWDL. | 76 |
| FIGURA D.20. TIPO DE RESPUESTA DEL SERVICIO. | 76 |
| FIGURA D.21. SELECCION DE CODIFICADOR Y EL PROTOCOLO DE MENSAJERIA. | 77 |
| FIGURA D.22. PROPIEDADES DELA VERSION DE HL7..... | 78 |
| FIGURA D.23. WSDL CREADA. | 79 |
| FIGURA D.24. WSDL BINDING..... | 79 |
| FIGURA D.25. VALIDAR WSDL..... | 80 |
| FIGURA D.26. PARTNERLINK DEL SERVICIO. | 81 |
| FIGURA D.27. PROCESO VACIO. | 82 |
| FIGURA D.28. RECIBIR REQUEST. | 82 |
| FIGURA D.29. RESPUESTA DEL PROCESO. | 83 |
| FIGURA D.30. AGREGAMOS UNA ASIGANCION..... | 84 |
| FIGURA D.31. SETEAMOS LOS VALRES DE LA ASIGANCION QUE SERÁN CODIFICADOS POR HL7..... | 84 |
| FIGURA D.32. CREAMOA APLICACION DE COMPOSICIÓN. | 86 |
| FIGURA D.33. NOMBRE DE LA APLICACION.FIGURA D.34..... | 86 |
| FIGURA D.35. AGREGAR MODULO JBI. | 87 |
| FIGURA D.36 IMPORTAR SERVICIO DE CODIFICADOR HL7..... | 87 |
| FIGURA D.37. INCLUIR SOPORTE PARA MODELADO..... | 91 |
| FIGURA D.38. ICLUIR SOPORTE PARA DEFINICION DE XML..... | 92 |
| FIGURA D.39. VERIFICACION DEL ENTORNO. | 93 |
| FIGURA D.40. ABRIR PROYECTO PLANTILLA. | 94 |
| FIGURA D.41. ENTORNO DE MAPERO HL7 V2 A V3..... | 95 |
| FIGURA D.42. MAPEO DE UN ATRIBUTO..... | 95 |
| FIGURA D.43. CREACION DE UN SERVICIO PARA HABILITAR LA TRASFORMACION. | 96 |
| FIGURA F.1. CASOS DE USO SISTEMA DE TELE-CONSULTA..... | 101 |
| FIGURA F.2. CASOS DE USO AUTORIA MEDICA..... | 105 |
| FIGURA F.3. MODELO CONCEPTUAL..... | 106 |



| | |
|---|------------|
| FIGURA F.4. DIAGRAMA DE SECUENCIA SERVICIO TELE CONSULTA. FLUJO COMUN. | 107 |
| FIGURA F.5. DIAGRAMA DE SECUENCIA SERVICIO DE TELECONSULTA. PARAMETRIZADO | 108 |
| FIGURA F.6. DIAGRAMA DE SECUENCIA DE AUDITORIA. | 108 |
| FIGURA F.7. DIAGRAMA DE PAQUETES..... | 109 |
| FIGURA F.8. MODELO FISICO DEL PROTOTIPO SOLUCIÓN. | 110 |
| FIGURA F.9. ADMINISTRACIÓN DE SERVIDORES NETBEANS..... | 111 |
| FIGURA F.10. SELECCIÓN DE GLASSFISH 2.X..... | 111 |
| FIGURA F.11. COFIGURACIÓN DE GLASSFISH EN NETBEANS 7.0 | 112 |
| FIGURA F.12. CONFIGURACIÓN DEL DOMINIO. | 112 |
| FIGURA F.13. SESSION DE ADMINISTRACIÓN. | 113 |
| FIGURA F.14. CREAR UN MÓDULO EJB. | 113 |
| FIGURA F.15. MODELO RELACIONAL..... | 114 |
| FIGURA F.16. SELECCION DE JDBC RESOURCE. | 115 |
| FIGURA F.17. SELECCIÓN DEL POOL DE CONEXIONES EXISTENTE..... | 116 |
| FIGURA F.18. SELECCIONAR PERSISTENCIA MEDIANTE JPA. | 117 |
| FIGURA F.19. SELECCION DE LA FUENTE DE DATOS..... | 118 |
| FIGURA F.20. SELECCION DE LAS CLASES Y EL PAQUETE AL QUE PERTENECEN. | 119 |
| FIGURA F.21. CREANDO UNIDAD DE PERSISTENCIA. | 120 |
| FIGURA F.22. MODELO OBJETO RELACIONAL JAVA. | 120 |
| FIGURA F.23. MODELO DE DATOS RELACIONAL..... | 120 |
| FIGURA F.24. SELECCIONAR ENTIDADES. | 121 |
| FIGURA F.25. CREAR EL CONTROLADORES A LAS ENTIDADES COMO EJBS..... | 122 |
| FIGURA F.26. ENTIDADES COMO EJB DE PERSISTENCIA. | 123 |
| FIGURA F.27. FACHADAS PARA ACCESO LOCAL Y REMOTO. | 123 |
| FIGURA F.28. CREAR UN BEAN DE SESION..... | 124 |
| FIGURA F.29. CREAR UN BEAN DE SESSION ADMINISTRAR OBSERVACIONES..... | 125 |
| FIGURA F.30. EJB ADMINISTRAR OBSERVACIONES. | 126 |
| FIGURA F.31. BEAN DE SESIÓN PARA ADMINISTRAR OBSERVACIONES. | 126 |
| FIGURA F.32. AGREGAR NUEVO SERVICIO WEB..... | 127 |
| FIGURA F.33. CREAR UN WEB SERVICE DESDE UN EJB DE SESIÓN..... | 127 |
| FIGURA F.34. WEB SERVICE DESDE UN EJB DE SESSION..... | 128 |
| FIGURA G.1. DIAGRAMA DE ARQUITECTURA CLUSTES..... | 129 |
| FIGURA H.1 MODELO CONCEPTUAL DE UN CANAL O INTERFAZ HL7 EN MIRTH. | 137 |
| FIGURA H.2. ICONO Y MENÚ DE OPCIONES DE MIRTH SERVER MANAGER EN EL ÁREA DE NOTIFICACIÓN DE WINDOWS. | 138 |
| FIGURA H.3. MIRTH CONNECT MANAGER..... | 138 |
| FIGURA H.4: VALIDACIÓN MIRTH ADMINISTRATOR..... | 139 |
| FIGURA H.5. VENTANA PRINCIPAL DE MIRTH (DASHBOARD)..... | 140 |
| FIGURA H.6. VENTANA PARA EDITAR UN CANAL (EDIT CHANNEL)..... | 141 |
| FIGURA H.7. VENTANA DE ERROR..... | 142 |
| FIGURA H.8. VENTANA DE CONFIGURACIÓN DEL CANAL. PESTAÑA SOURCE. | 143 |
| FIGURA H.9. VENTANA DE CONFIGURACIÓN DEL TRANSFORMADOR. | 144 |
| FIGURA H.10. VENTANA DE CONFIGURACIÓN DEL TRANSFORMADOR. PESTAÑA MESSAGE TEMPLATES. | 145 |
| FIGURA H.11. VENTANA DE CONFIGURACIÓN DEL TRANSFORMADOR. PESTAÑA MESSAGE TREES. | 147 |
| FIGURA H.12. ESPECIFICACIÓN DE LAS VARIABLES DE MAPEO EN LOS PASOS DE TRANSFORMACIÓN | 148 |
| FIGURA H.13. VENTANA DE CONFIGURACIÓN DEL TRANSFORMADOR. MAPEO DE VARIABLES EN LOS PASOS DE TRASNFORMADOR. | 148 |



FIGURA H.14. ESPECIFICACIÓN DEL MAPEO DE SEGMENTOS EN LOS PASOS DE TRANSFORMACIÓN..... 149

FIGURA H.15. VENTANA DE CONFIGURACIÓN DEL TRANSFORMADOR. MAPEO DE SEGMENTOS DE MENSAJE EN LOS PASOS DE TRANSFORMACIÓN (TRANSFORMER STEPS)..... 149

FIGURA H.16. MAPEO DE LOS SEGMENTOS DE MENSAJE..... 150

FIGURA H.17. VENTANA DE CONFIGURACIÓN DEL TIPO DE CONECTOR DATABASE WRITER. 151

FIGURA H.18. VENTANA PARA PUBLICACIÓN Y DESPLIEGUE DEL CANAL. 152

FIGURA H.19. CREAR CANAL SOAP..... 153

FIGURA H.20. METODO ENCAPSULADO DE RESPUESTA. 153

FIGURA H.21. VALIDACIÓN DEL CANAL. 154



ÍNDICE DE TABLAS

| | |
|---|-----|
| TABLA A.1. OPERACIONES Y CONTEXTOS DE LÓGICA. | 14 |
| TABLA A.2. MAPA DE FUNCIONES DE NEGOCIO HSLV..... | 17 |
| TABLA 3. LOGICA DE APLICACION DE SERVICIOS CANDIDATOS..... | 18 |
| TABLA 4. AGRUPACION DE APLICACION DE SERVICIOS CANDIDATOS | 19 |
| TABLA 5. ENTRADAS Y SALIDAS DE LOS SERVICIOS DE APLICACIÓN. | 26 |
| TABLA 6. ESENARIOS DE INTERACCIÓN. | 31 |
| TABLA 7. PRUEBAS DE ESCALABILIDAD. | 35 |
| TABLA 8. NIVELES DE SERVICIO ACEPTABLES. | 35 |
| TABLA 9. DISTRIBUCIÓN DE SERVICIOS. | 36 |
| TABLA B.1. CARACTERÍSTICAS DE OPENESB..... | 37 |
| TABLA B.2. CARACTERÍSTICAS DE CHAINBUILDER..... | 38 |
| TABLA B.3. CARACTERÍSTICAS DE FUSE ESB..... | 38 |
| TABLA B.4. CARACTERÍSTICAS DE PETALS ESB. | 39 |
| TABLA B.5. CARACTERÍSTICAS DE JBOSS ESB. | 40 |
| TABLA B.6. CARACTERÍSTICAS DE APACHE SYNAPSE. | 40 |
| TABLA B.7. CARACTERÍSTICAS DE SERVICEMIX. | 41 |
| TABLA B.8. CARACTERÍSTICAS DE WSO2 ESB. | 41 |
| TABLA B.9. CARACTERÍSTICAS DE MULE ESB..... | 42 |
| TABLA B.10. CARACTERÍSTICAS JEEESB. | 43 |
| TABLA B.11. CARACTERÍSTICAS DE CELTIX. | 43 |
| TABLA B.12. COMPARACIÓN DE LOS ESB. | 44 |
| TABLA 22. CASO DE USO INICIAR SESIÓN. | 101 |
| TABLA 23. CASO DE USO VALIDAR ACCESO..... | 102 |
| TABLA 24. CASO DE USO CONSULTAR INFORMACIÓN CLINICA. | 102 |
| TABLA 25. CASO DE USO CONSULTAR EPICRISIS..... | 102 |
| TABLA 26. CASO DE USO CONSULTAR HISTORA CLINICA..... | 103 |
| TABLA 27. CASO DE USO CONSULTAR EXAMENES DE LABORATORIO..... | 103 |
| TABLA 28. COSULTAR IMAGENES DIAGNOSTICAS..... | 104 |
| TABLA 29. CASO DE USO AUDITAR INGRESO. | 105 |
| TABLA 30. CASO DE USO CONSULTAR FACTURA..... | 105 |
| TABLA 31. PRUEBAS DE RENDIMIENTO. | 129 |
| TABLA 32. PRUEBAS DE CARGA. | 133 |
| TABLA 33. RESULTADOS DE PRUEBAS DE CARGA | 133 |
| TABLA 34. PRUEBAS DE STRESS..... | 134 |
| TABLA 35 RESULTADOS DE PRUBAS DE ESTRES PARA CLUSTER CON UN NODO..... | 134 |



A DISEÑO DE LA ARQUITECTURA BAJO EL MODELO DE REFERENCIA SOA.

La aplicación de esta metodología está orientada a construir una SOA que soporte los procesos de negocio del HSLV; para este caso de estudio, se emplea la metodología para dos procesos de negocio en particular que se describen a continuación:

- **Auditoría:** permite hacer una verificación de los servicios prestados por el HSLV y sus valores facturados.
- **Tele-consulta:** permite a personal médico especializado de otras entidades conectadas a través de la Red Nacional de Tecnología Avanzada (RENATA) el acceso a la información clínica de los pacientes del HSLV.

Para un caso general, debe desarrollarse todo el proceso metodológico considerando la totalidad de los procesos de negocio de la IPSS.

Se describen las fases correspondientes a la metodología de Orientación a Servicios.

A.1. ANÁLISIS ORIENTADO A SERVICIOS.

De esta etapa de la metodología se deben tener en cuenta los siguientes ítems.

- Hacer un estudio de cómo los requerimientos de automatización se pueden representar a través de la orientación a servicios.
- Dimensionar los servicios que se requieren para su posterior construcción y la lógica que debe encapsular cada uno de estos.
- Definir Contextos lógicos para agrupar los servicios candidatos y su potencial de reutilización.
- Establecer los límites de los servicios para evitar que se traslapen o redunden operaciones ya encapsuladas.
- Establecer un modelo preliminar de composición de servicios conocidos.

A.1.1. DEFINIR REQUERIMIENTOS DE NEGOCIO.

Los requerimientos de negocio se recolectan y se documentan de forma clara y concisa. Esta documentación será el punto de inicio del Modelado de Servicios candidatos (Paso 1.3).

A.1.1.1. Proceso de negocio Auditoria.

Descripción: los servicios asistenciales prestados por el HSLV luego de pasar por el proceso de facturación, deben ser verificados por personal calificado de manera que confronte los datos clínicos del servicio con el valor asociado en la factura a cada uno de ellos.

Requerimientos:



- La información disponible debe estar habilitada exclusivamente para personal autorizado. El personal debe identificarse con un nombre de usuario y contraseña.
- Debido a las restricciones legales respecto a la privacidad de la información contenida en la historia clínica de los pacientes, el auditor debe poder acceder a un resumen con información relevante de los servicios prestados en un ingreso a determinado paciente; esto se le conoce como epicrisis¹.
- Cada ingreso de un determinado paciente tiene asociada una factura que contiene en detalle los servicios y el costo de cada uno de estos. El auditor debe tener acceso a esta información para corroborarlos con la epicrisis.
- En caso de hallarse alguna inconsistencia, el auditor debe poder agregar sus observaciones relacionadas con determinado servicio facturado.

A.1.1.2. Proceso de Negocio Tele-consulta.

Descripción: el objetivo es permitir el acceso desde cualquier centro de atención de salud que pertenezca a RENATA a la información contenida en la historia clínica de un paciente en particular.

Requerimientos:

- El personal médico de una institución debe acceder a la información disponible luego de validar su autenticación mediante un nombre de usuario y contraseña.
- La epicrisis debe estar disponible para ser consultada por el personal médico autorizado.
- En caso de requerirse más información, se debe hacer una consulta a la historia clínica detallada del paciente.
- Para determinados casos, donde la información debe ser más específica, debe poderse acceder a los resultados de los exámenes médicos practicados a un determinado paciente y/o a las imágenes diagnósticas.

A.1.2. IDENTIFICAR SISTEMAS AUTOMATIZADOS.

Identificar la lógica de aplicaciones existentes que automatizan algunos de los requerimientos de negocio. Esto permite dimensionar el potencial de los procesos existentes.

El sistema de Información Dinámica Asistencial permite al personal médico el almacenamiento de información clínica de los servicios prestados en el HSLV; Por lo tanto, la información clínica ya se encuentra en las bases de datos del HSLV. La aplicación Datalab y Carestream proveen la información de los exámenes de laboratorio e imágenes diagnósticas respectivamente.

El sistema dinámica gerencial es el que permite el proceso de facturación de servicios.

¹ Resumen de los servicios asistenciales relacionados con el ingreso de un paciente a la IPSS.



Con el escenario descrito anteriormente, se puede afirmar que la información básica para realizar los procesos y servicios se encuentra disponible en los sistemas automatizados que se encuentran en operación actualmente. Sin embargo, no existen las estructuras de almacenamiento que permitan el proceso de auditoría.

A.1.3. MODELADO DE SERVICIOS CANDIDATOS.

Las operaciones de servicio candidatos son identificadas y agrupadas dentro de un contexto de lógica. Estos grupos toman forma como servicios candidatos que después serán ensamblados dentro de un modelo de composición tentativo que representa la lógica combinada de las aplicaciones planeadas orientadas a servicios.

A.1.3.1 Descomponer procesos de negocio.

Tomar los procesos de negocio documentados y descomponerlos en pasos de procesos granulares. Es importante que la lógica del flujo de trabajo de los procesos sea descompuesta dentro de representaciones lo más granulares de pasos de procesamiento, el cual diferirá del nivel de granularidad con el cual el proceso fue documentado.

Siguiendo la propuesta para el modelado Orientado a Servicios se presentan la descomposición de los procesos de negocio en procesos granulares.

Los procesos granulares que componen el proceso Auditoria son:

- Solicitar el servicio
- Verificar que el usuario tenga permisos para consumir el servicio
- Enviar solicitud de epicrisis y factura.
- Obtener las epicrisis
- Obtener la factura asociada.
- Guardar una Observación
- Borrar una observación
- Listas observaciones
- Modificar Observaciones
- Consultar Observaciones.

Considerando que una solicitud a un proceso distribuido esta adjunta a una respuesta; en el casos de los procesos granulares de envío de solicitud los podemos encapsular en un solo servicio como lo es Obtener la respuesta a la solicitud. De esta forma los procesos de negocio son casi los mismos procesos que de identifican en las figuras anteriores.

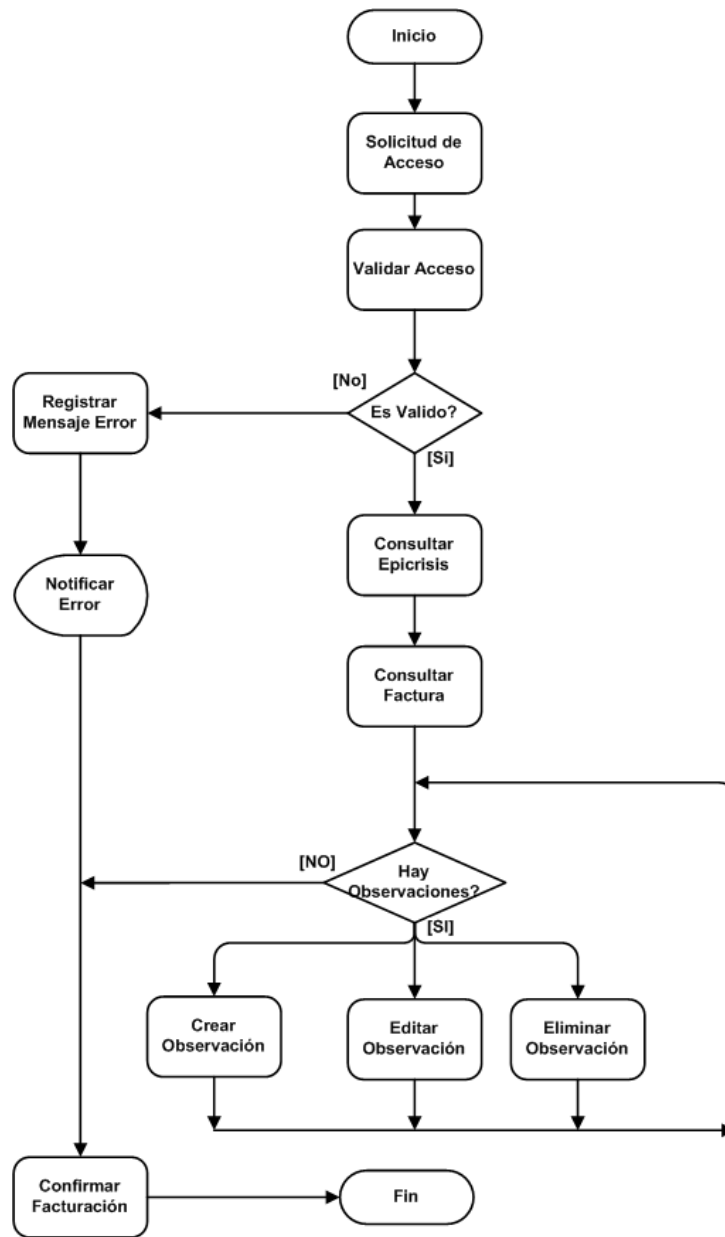


Figura A.1. Proceso Auditoria

Los procesos granulares que componen el proceso Tele-consulta son:

- Solicitar el Servicio
- Verificar que el usuario tenga permisos para consumir el servicio
- Enviar solicitud de epicrisis.
- Obtener epicrisis
- Enviar solicitud de historia Clínica
- Obtener datos de la Historia Clínica
- Enviar solicitud de consulta de Exámenes
- Obtener Resultados de Exámenes de Laboratorio



- Obtener imágenes diagnósticas.

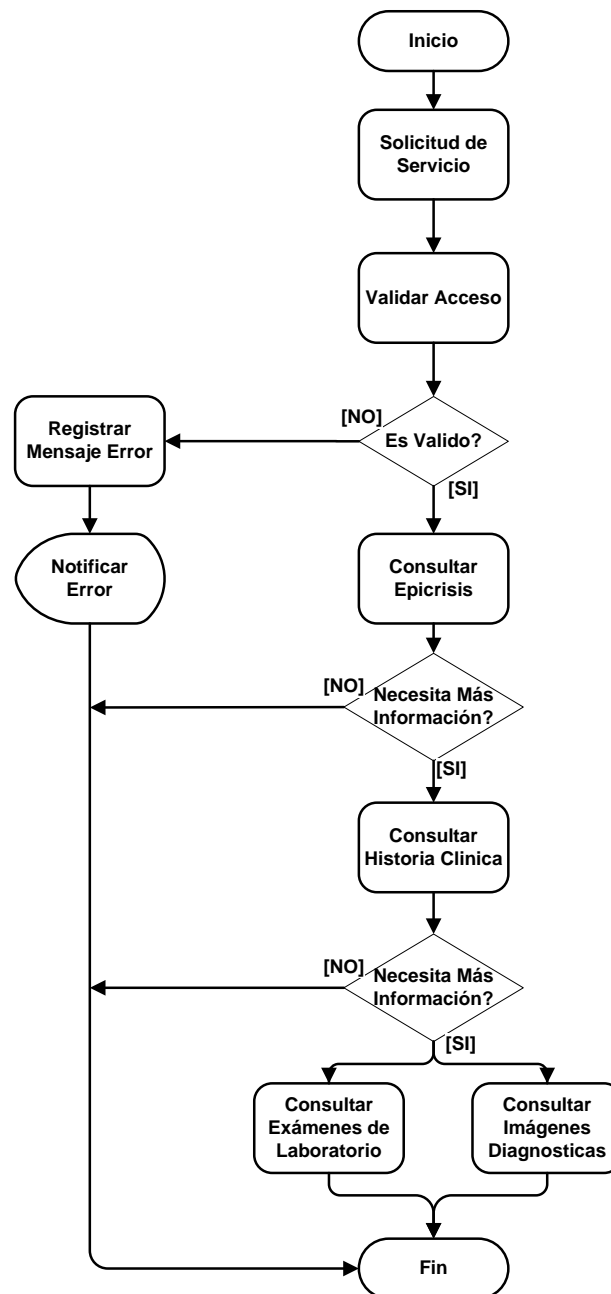


Figura A.2. Proceso Tele-consulta.

A.1.3.2. Identificar Operaciones de Servicios de Negocio Candidatas.

Algunos de los pasos u operaciones que se encuentran dentro de la lógica encapsulada en un servicio pueden no necesariamente pertenecer exclusivamente a este ámbito.



Se decide omitir este paso dado la simplicidad de las operaciones que componen el proceso de negocio.

El proceso de negocio Tele consulta aún se encuentra entre las necesidades actuales del HSLV debido a que no se está dando soporte a este proceso. Las operaciones descritas anteriormente no están siendo soportadas por ninguno de los sistemas de información existentes en el HSLV.

A.1.3.3. Extraer la Lógica de Orquestación.

En esta sub fase del análisis se deben estudiar aspectos como lo son:

- *Reglas de negocio.*
- *Lógica condicional.*
- *Lógica de excepciones.*
- *Lógica de secuencia.*

Proceso Auditoria:

- Se solicita acceso al proceso de negocio.
- Si se valida el acceso al proceso.
- Si la validación es correcta se puede continuar con el resto del proceso.
- En caso de no se pueda continuar con el proceso se ejecuta la función de registrar el acceso fallido.
- Si se puede continuar con el proceso el sistema inicia la función de consultar la epicrisis y la factura asociada.
- Si el usuario auditor nota de que debe hacer una anotación puede tomar algunas de las opciones de administración de observaciones que son:
 - Crear una observación.
 - Editar una observación.
 - Eliminar una observación.
 - Borrar una observación.
- El auditor debe volver a estar en condiciones de decidir si debe o no hacer una observación a los valores facturados o no.
- Si el usuario auditor no estima conveniente hacer observaciones simplemente debe confirmar que la factura ha sido revisada.

Proceso de Tele-consulta:

- Se solicita acceso al proceso de negocio.
- Si se valida el acceso al proceso.
- Si la validación es correcta se puede continuar con el resto del proceso.
- Ahora se puede consultar la epicrisis de un paciente.
- El medico analiza la necesidad de consultar más información.
- En caso de no requerir más información finaliza el proceso.
- En caso de solicitar más información puede consultar la historia clínica.



- Ahora el medico analiza la información y puede determinar si desea o no consultar otra información.
- En el caso de no requerir más información, termina el proceso.
- Si se estima la necesidad de consultar información adicional el medico puede ejecutar procesos que se mencionan a continuación:
 - Consultar exámenes de Laboratorio.
 - Consultar Imágenes Diagnosticas.

A.1.3.4. Crear Servicios Candidatos.

Revisar los pasos de procesamiento que faltan y determinar uno o más contextos lógicos con los cuales los pasos puedan ser agrupados. Cada contexto representa un servicio candidato. El contexto final dependerá de los tipos de servicios de negocio que se haya escogido para construir.

La propuesta de este ejercicio es establecer el conjunto de contextos requeridos.

También se recomienda que los servicios candidatos de negocio centrados en entidades estén equipados con operaciones candidatas adicionales que faciliten su futura reutilización. Además, lo importante de este paso puede ser expandido al incluir un análisis de operaciones candidatas de servicio adicional que no requieren del proceso de negocio normal, pero agregar un conjunto de servicios con operaciones reutilizables.

La Tabla A.1. se presentan las operaciones y el contexto de lógica asociado a los dos procesos de negocio.

Tabla A.1. Operaciones y Contextos de Lógica.

| | Operaciones | Contexto de Lógica |
|------------------------------|-------------------------------------|-------------------------------------|
| Proceso de Auditoria. | Validar Acceso al servicio. | Validar Acceso |
| | Crear Observación. | Administración de Observaciones |
| | Editar Observación. | Administración de Observaciones |
| | Eliminar observación. | Administración de Observaciones |
| | Consultar Observación. | Administración de Observaciones |
| | Consultar epicrisis. | Consultar Historia Clínica. |
| | Consultar Factura. | Consulta de factura. |
| | Error de acceso a servicio. | Gestión de Acceso no Válido. |
| | Registro de errores. | Gestión de Acceso no Válido. |
| Proceso Tele-Consulta | Validar Usuario | Validar Acceso. |
| | Consultar epicrisis | Consultar Historia Clínica. |
| | Consultar historia clínica | Consultar Historia Clínica. |
| | Consultar resultados de laboratorio | Consultar resultado de Laboratorio. |
| | Consultar Imágenes Diagnósticas. | Imágenes diagnósticas. |
| | Error de acceso a servicio. | Administración de Excepciones |
| | Registro de errores. | Administración de Excepciones |



Figura A.3. Servicios Candidatos.

A.1.3.6. Identificar Composiciones de Servicio.

Identificar un conjunto de los escenarios más comunes que pueden tomar lugar dentro de los límites de los procesos de negocio. Para cada uno de estos escenarios hay que seguir los pasos de procesamiento:

Esto da una Buena idea de cómo apropiar el agrupamiento de los pasos de procesos.

Demuestra la relación potencial entre orquestación y los niveles de servicios de negocio.

Identifica el potencial de composición de servicios.

Él pone a la vista cualquier lógica de flujo de trabajo o paso de procesamiento faltante.

Garantizar que como parte de los escenarios escogidos incluir condiciones de fallas que involucran la lógica para el manejo de excepciones. Todos los niveles de servicio establecidos en este punto están todavía en forma preliminar y pueden estar sujetos a revisiones durante el proceso de diseño.

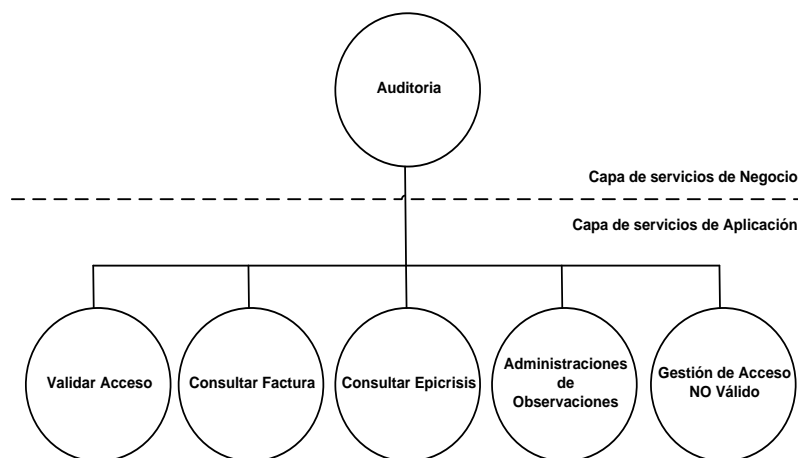


Figura A.4. Representación de la composición de proceso Auditoria.

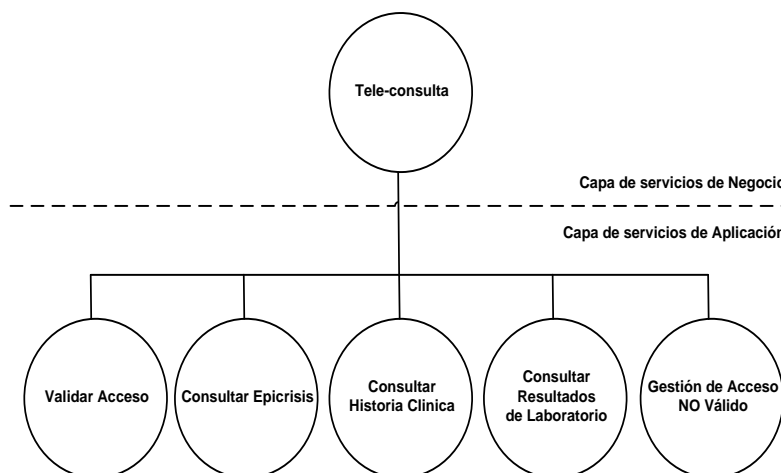


Figura A.5. Representación de la composición de proceso Tele-consulta.

A.1.3.7. Revisar Operación de Grupo.

Basado en los resultado de la composición, revisar el agrupamiento de los procesos de negocio y revisar la organización de operaciones de servicio candidatos como sea necesario.



Considerando las recomendaciones para el establecimiento de una SOA según la guía propuesta en el proyecto HSSP (HSSP, 2009) para empresas prestadoras de servicios de salud (Paso 1, arquitectura empresarial), que nos permite definir las líneas de negocio (Paso 2) y la identificación del grupo de servicios candidatos (Paso 3), y de esta forma establecer el mapa de funciones de negocio para el HSLV como se muestra en la Tabla A.2.

Tabla A.2. Mapa de funciones de negocio HSLV.

| LÍNEAS DE NEGOCIO | SERVICIOS ÚNICOS DE SALUD | | | | | | SERVICIOS DE NEGOCIO | | | | | | SERVICIOS DE INFRAESTRUCTURA | | | | | |
|-----------------------------------|---------------------------|-----------------|--------------------------|----------------------|-------------------|-------------------------------|--------------------------------------|-----------------------------|----------------------------|---------------|--------------|------------------------|------------------------------|--------------------------------|-----|----------------|--------------------------|--------------------|
| | Orden de Ingreso | Orden de Salida | Valoración de Pacientes. | Datos de Laboratorio | Datos de Farmacia | Registro Clínico Electrónico. | Administración de Alertas y Eventos. | Índice Maestro de Personas. | Terminología de Servicios. | Demográficos. | Facturación. | Programación de Citas. | Servicio de Auditoría. | Administración de Excepciones. | BPM | Autenticación. | Directorio de Servicios. | Control de Acceso. |
| Farmacia. | X | X | X | | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| Laboratorio. | X | X | X | X | | X | X | X | X | X | X | X | X | X | X | X | X | X |
| Administración de Pacientes. | X | X | | | | | X | X | | X | X | | X | X | | X | X | X |
| Adm. Orden de Ingreso. | X | X | | | | X | X | X | X | X | X | X | X | X | X | X | X | X |
| Programación de Citas. | X | | | | | | X | X | | X | | X | X | X | X | X | X | X |
| Administración de Atención. | | | | | | | X | X | | X | X | X | X | X | X | X | | X |
| Administración de Remisiones. | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| Enfermería. | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| Emergencias. | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| Cirugías. | X | X | X | X | X | X | X | X | X | X | X | X | X | X | | X | X | X |
| Salud Mental. | | X | | X | X | X | X | X | | X | X | | X | X | X | X | | X |
| Facturación. | | X | | | | X | X | X | X | X | X | X | X | X | X | X | X | X |
| Cartera. | | | X | X | X | X | X | | X | X | | | X | X | | X | | X |
| Contabilidad. | | X | | | X | | X | | | | | X | X | X | | X | X | X |
| Cartera. | X | X | | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| Imágenes Diagnósticas. | X | X | X | | X | X | X | X | X | X | X | | X | X | X | X | | X |
| Gestión de la calidad. | X | X | X | X | | X | X | X | X | X | X | X | X | X | X | X | | X |
| Administración para la Nutrición. | | | | | | | X | X | | X | X | | X | X | | X | X | X |

A.1.3.8. Analizar requerimientos de procesamiento.

Este paso es opcional y más cómodo para procesos de negocio complejos y grandes entornos orientados a servicios. Requiere un estudio cercano al nivel bajo de requerimientos de procesamiento de todos los servicios candidatos para abstraer cada servicio candidato centrado en la tecnología desde esta vista que completará el nivel de servicios de aplicación. Para lograrlo, cada paso de procesamiento identificado es requerido para dirigir un pequeño análisis.

Específicamente lo que se debe determinar es:



Que lógica de aplicación necesita ser ejecutada para procesar la acción descrita por la operación candidata.

Cuando la lógica de aplicación requerida ya exista o cuando necesite ser nuevamente desarrollada.

Cuando la lógica de aplicación requerida extiende los límites de la aplicación. Es decir, es más de un sistema requerido para completar la acción.

Tabla 3. logica de aplicacion de servicios candidatos.

| Servicio Candidato | Lógica de Aplicación |
|---|--|
| Validar Acceso. | Recibe los datos para identificar el consumidor del servicio así como la ubicación del mismo. |
| | Identifica si se tiene acceso o no al proceso requerido. |
| | Retorna un mensaje de error y dispara el registro de errores en caso de que el acceso no sea válido. |
| | En caso de que el acceso sea válido, habilita el flujo normal del proceso. |
| Administración de observaciones. | Recibe un mensaje que contiene el tipo de operación a ejecutar: eliminar, crear, consultar y edita. |
| | Como resultado se obtiene el estado actual de las observaciones registradas. |
| Consultar Historia Clínica. | Recibe una solicitud que contiene el identificador de la consulta clínica requerida. |
| | Accede a la información de la consulta clínica. |
| | Retorno de la información en el formato adecuado. |
| Consultar Factura. | Recibe una solicitud del consumidor del servicio con el identificador de la factura deseada. |
| | Accede a la información de la factura |
| | Retorna la información de la factura en formato adecuado. |
| Gestión de acceso no valido. | Recibe la información del servicio solicitado y el consumidor que lo requiere. |
| | Almacena la información de negación del acceso. |
| Consultar Epicrisis. | Recibe una solicitud que contiene el identificador de la epicrisis requerida. |
| | Accede a la información de la consulta de epicrisis. |
| | Retorno de la información en el formato adecuado. |
| Consultar resultados de Laboratorio. | Recibe la información que identifica los resultados solicitados. |
| | Accede a los resultados solicitados. |
| | Retorna los resultados en formato requerido. |
| Consultar imágenes diagnósticas. | Recibe la información que identifica las imágenes solicitadas. |
| | Accede a los recursos solicitados. |
| | Retorna los resultados en formato requerido. |

A.1.3.9. Identificar Operaciones de Servicio de Aplicación.



Descomponer cada requerimiento de procesamiento de lógica de aplicación dentro de una serie de pasos. Ser explícitos sobre cómo se llamarán estos pasos, de tal manera que ellos referencien el funcionamiento de su desempeño. Idealmente, no se debería referenciar el paso del proceso de negocio para el cual esta función está siendo identificada.

Dada la simplicidad de los procesos, las operaciones de servicio y requerimientos de procesamiento damos paso al siguiente ítem propuesto por la metodología.

A.1.3.10. Crear los Servicios Candidatos de Aplicación.

Agrupar estos pasos de procesamiento de acuerdo a un contexto predefinido. Con servicios de aplicación candidatos, el contexto primario es una relación lógica entre operaciones candidatas. Esta relación puede estar basada en algún número de factores, incluyendo:

- *Asociación con un sistema legado específico.*
- *Asociación con uno o más componentes de solución.*
- *Agrupar la lógica de acuerdo al tipo de función.*

Muchos otros activos son construidos en un servicio candidato y esta sujetos al proceso de diseño orientado a servicios.

Dado que los sistemas legados no están habilitados para ser accedidos solo por los proveedores de las soluciones, se omite esta asociación, dado que se asume solo los sistemas legados al nivel de almacenamiento (repositorios de datos).

Tabla 4. Agrupacion de aplicacion de servicios candidatos

| Servicio Candidato | Agrupación de aplicación. |
|---|----------------------------------|
| Validar Acceso. | Control de acceso. |
| Consultar Historia Clínica. | Consulta Información. |
| Consultar Factura. | Consultar Información. |
| Gestión de acceso no valido. | Control de acceso |
| Consultar Epicrisis. | Consulta de información. |
| Consultar resultados de Laboratorio. | Consulta de información. |
| Consultar imágenes diagnósticas. | Consulta de información. |

A.2. DISEÑO ORIENTADO A SERVICIOS

Es el proceso por el cual, los servicios son derivados de la lógica de los servicios candidatos para después ser ensamblados dentro de una composición abstracta que implementa un proceso de negocio.

El objetivo de esta fase es determinar cómo pueden ser las definiciones de interfaces de servicio físicas ser derivadas de los servicios candidatos modelados durante la fase de análisis orientado a servicios, para lo cual.



- *Cuál característica de SOA deseamos realizar y soportar.*
- *Cual estándar y extensión de la industria será requerida por medio de nuestra SOA para implementar los diseños de servicios planeados y características de SOA.*

Para dirigir estas preguntas, el proceso de diseño involucra más que un análisis. El centro estará sobre los factores del entorno y los estándares de diseño que dan forma a los servicios.

Objetivos de un diseño orientado a servicios:

- Determinar el núcleo del conjunto de extensiones arquitectónicas.
- Establecer los límites de la arquitectura.
- Identificar los estándares de diseño requeridos.
- Definir el diseño de las interfaces de servicio abstraídas.
- Identificar la composición de servicios potencial.
- Explorar el soporte para las características de SOA.
- Corresponde con el diseño del flujo de trabajo solución.

A.2.1. COMPOSICIÓN DE SOA.

En esta fase se concentra en obtener el diseño de los servicios de negocios compuestos.

A.2.1.1. Escoger los Niveles de Servicio.

La composición de SOA requiere que se decida una configuración de diseño para los niveles de servicio que comprometerán y estandarizarán la representación lógica dentro de la arquitectura. Se hace un estudio de los niveles de servicio candidatos producidos durante la fases de análisis orientado a servicios y la exploración de niveles de servicio y los escenarios de configuración de niveles de servicios.

Los niveles de referencia de la metodología son:

- Nivel de Servicios de aplicación.
- Nivel de Servicios de Negocio.
- Nivel de Servicios de Orquestación.

Se resalta la separación de niveles para establecer la flexibilidad necesaria que se requiere para la construcción de SOA. Esta separación facilita la definición de nuevos procesos y de otros ya existentes.

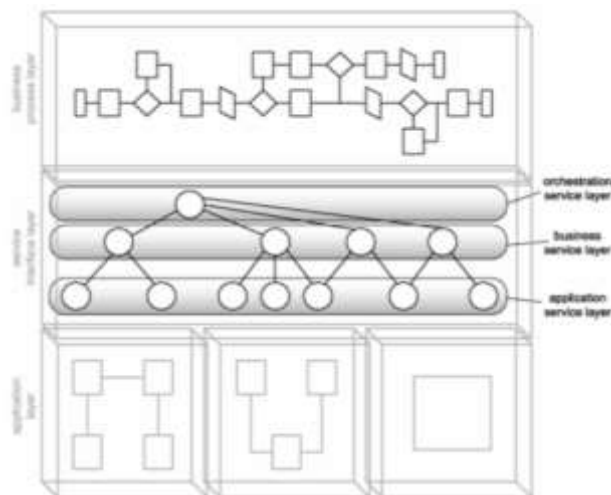


Figura A.6 Niveles de servicios.

Para el caso de la utilización de alguna tecnología de middleware, el nivel de servicios de orquestación estaría soportado por este, con la finalidad de administrar de forma fácil y rápida los servicios disponibles en el entorno empresarial.

A.2.1.2. Ubicación de los Estándares de Núcleo.

Es necesario establecer cuales estándares del núcleo podrían comprometer SOA y como ellos serían implementados para mejorar el soporte a las características y requerimientos de la solución orientada a servicios.

Lo que se obtiene en este paso de la arquitectura, es el establecimiento de los estándares requeridos; en la siguiente figura se muestran en concreto:

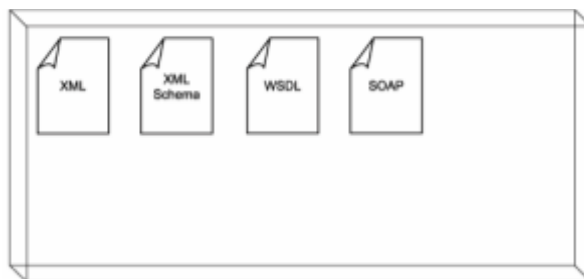


Figura A.7. Vistan de compiccion del nucleo de estandares para la SOA.

A.2.1.3 Seleccionar las Extensiones de SOA.

Determinar cuáles de las características de SOA deseamos soportar. Esto ayuda a decidir cuáles de las especificaciones WS- disponibles podrían llegar a ser parte del entorno orientado a servicios.*



Las seleccionadas se presentan en la siguiente figura:

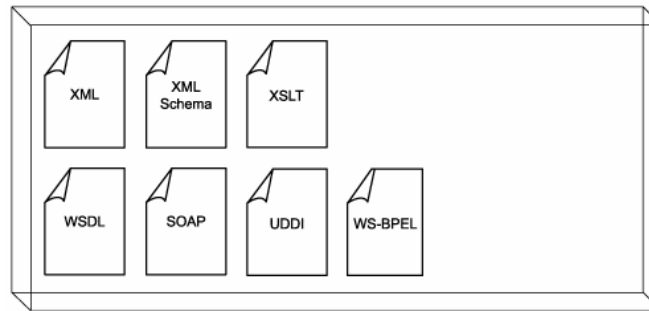


Figura A.8 Vista de los estenciones para la descripción de servicios.

A.2.2. DISEÑAR LOS SERVICIOS DE NEGOCIO.

Estos servicios representan el único nivel de servicio que está influenciado por los otros. Su propósito es precisamente representar las entidades de datos correspondientes dentro de un modelo de negocio de la organización (es decir son la fachada de un proceso de negocio). Estos servicios están contruidos para su reutilización por cualquier aplicación que necesite acceder o administrar información asociada con una entidad particular.

Existen en relación a otros niveles de servicio, esto es benéfico para diseñar los servicios de negocio centrados en entidades de forma prioritaria respecto a otros. Esto establece una abstracción de niveles de servicio alrededor de los cuales procesos y subniveles de lógica de aplicación pueden ser posicionada.

A.2.2.1. Revisar Servicios Existentes.

Cuando se crean servicios centrados en entidades, el esfuerzo de modelado resulta en los servicios candidatos que han de tomarse de algún servicio existente. Como los servicios candidatos tienen a consistir de operaciones candidatas relevantes para los requerimientos de negocio que forman la base del análisis orientado a servicios, siempre verificar el valor para garantizar que algunos o todas las funcionalidades de procesamiento representadas por una operación candidata no exista en otros servicios.

A.2.2.2. Definir el tipo de esquema de los mensajes.

Es útil empezar el diseño de una interfaz de servicio con una definición formal de los mensajes que el servicio requiere para procesar. Para lograr esto se necesita formalizar las estructuras de mensajes que son definidas dentro del área *Types* de WSDL.

A.2.2.3. Derivar y abstraer las interfaces de servicio.



Analizar el propósito de las operaciones de servicio candidatas y seguir estos pasos para definir una interfaz de servicio inicial:

- Confirmar que cada operación candidata genérica y reutilizable asegurando que la granularidad de la lógica encapsulada es apropiada.
- Estudiar las estructuras de datos definidas en el paso 2 y establecer un conjunto de nombre de operaciones.
- Crear los portType o interface dentro de la WSDL y relacionarlos con construcciones de operaciones que corresponda con las operaciones candidatas.

Formalizar la lista de valores de entradas y salidas requeridas para acomodar el procesamiento de cada lógica de operación. Esto se cumple definiendo la construcción de messages apropiados que referencie los tipos de esquemas XSD dentro de pequeñas partes de elementos.

A.2.2.4. Extender el diseño de servicios.

El proceso de modelado de servicios tiende a enfocarse en requerimientos de negocio evidentes. Mientras se promueve la reutilización a menudo falla en el proceso de diseño para garantizar que una cantidad suficiente de funcionalidades reutilizables. Para evitar esto se sugiere hacer un análisis especulativo de que otros tipos de características que podrían ofrecer los servicios dentro de un contexto funcional predefinido. Para implementar nuevas funcionalidades

- *Agregar nuevas operaciones.*
- *Agregar parámetros a operaciones existentes.*

Agregar operaciones es una forma de proveer funcionalidades asociadas evidentes a cada entidad. El conjunto básico sería:

- *Obtener algo.*
- *Actualizar algo.*
- *Agregar algo.*
- *Eliminar algo.*

Estableciendo estas operaciones estándares se construye un nivel de interoperabilidad consistente dentro del nivel de servicios de negocio, facilitando la reutilización y composición.

La aplicación de este diseño da como resultado la siguiente descripción de operaciones de los servicios de negocio.

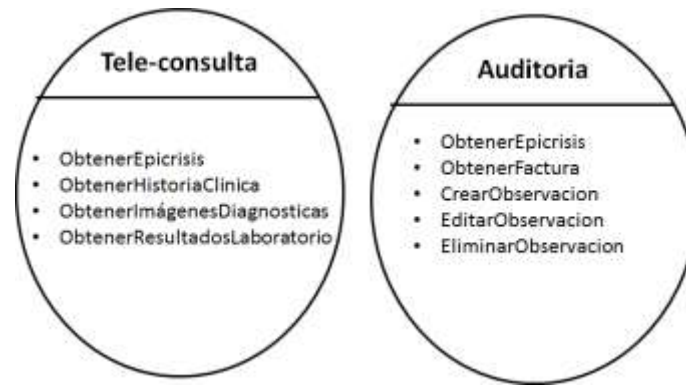


Figura A.9. Servicios de Negocio y sus operaciones.

La anterior figura describe los dos procesos de negocio y las operaciones de los mismos luego de aplicar los pasos descritos hasta la extensión de los servicios de negocio.

A.2.2.5. Identificar el procesamiento requerido.

Mientras el proceso de modelado de servicios desde el análisis orientado a servicios puede haber identificado algunos servicios de aplicación principales, estos no tienen que haber sido definidos en su totalidad.

Ahora se tiene un diseño real para estos nuevos servicios de negocio, se puede estudiar los requerimientos de procesamiento de cada una de sus operaciones más detalladamente. Haciendo esto, se podría determinar **si servicios de aplicación adicionales se necesita para transportar cada pieza de funcionalidad expuesta**. Si se encuentra la necesidad de nuevos servicios de aplicación, se tendrá que determinar si estos ya existen, o si ellos deben ser agregados a la lista de servicios que serán expuestos como parte de esta solución.

Por las características del servicio de auditoria se puede hacer una representación de este en las tres capas dado que el servicio de tele-consulta solo aborda la pasa de servicios de negocio.

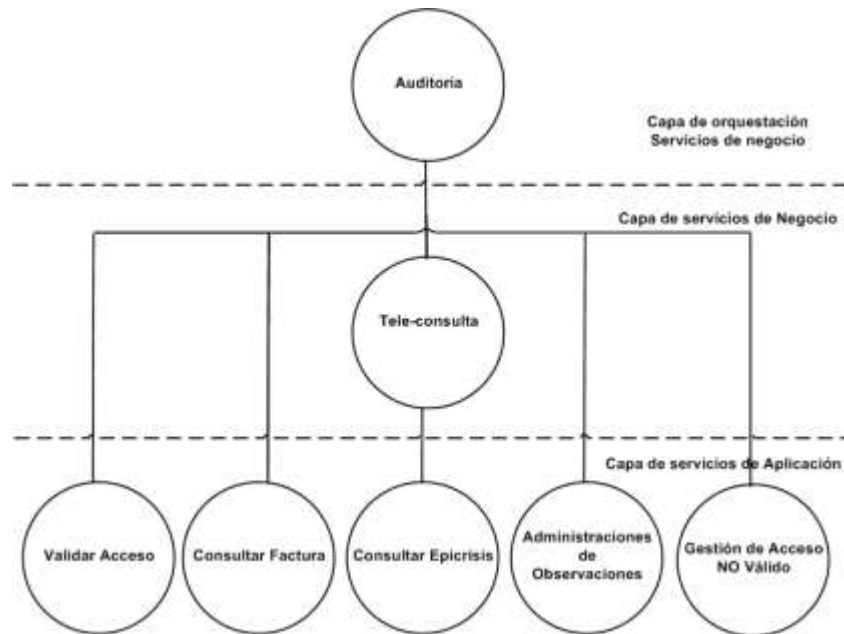


Figura A.10. Jerarquía de composición del servicio de negocio de Auditoría.

La Figura A.10 muestra la jerarquía de composición del servicio de auditoría que está compuesto por el servicio de negocio tele-consulta. No se despliega la figura de jerarquía del servicio de negocio tele-consulta porque no está compuesto por algún servicio de negocio, por tanto sería como el mostrado en la Figura A.5.

A.2.3. DISEÑAR LOS SERVICIOS DE APLICACIÓN.

Los servicios de aplicación son las herramientas de trabajo de SOA. Representan el subnivel mas bajo del nivel de servicios de composición, responsable de transportar cualquiera de las demandas de procesamiento dictadas por los niveles de orquestación y negocio.

El diseño de servicios de aplicación no requiere un nivel experto para el análisis de negocio. Es una abstracción de un entorno técnico de la organización, mejor definido por aquellos quienes entienden más esos entornos.

Tanto los servicios de aplicación como los servicios de negocio establecen lógica de servicios reutilizables y dependen de controladores mayores para componerlos dentro de tareas específicas de procesos de negocio.

No hay un paso en el cual se definan los requerimientos de procesamiento de todo el proceso de negocio en los servicios de aplicación. Es por esto, que estos servicios son los responsables para la implementación de los detalles de procesamiento requeridos para acarrear la lógica de negocio de sus servicios de negocio parentales.

A.2.3.1. Revisión de servicios existentes.



Es importante que en los servicios de aplicación (que son los más reutilizables en la arquitectura) asegurar que la funcionalidad sea específica, es decir que realice una operación particular. Debido a que estos servicios proveen tales funcionalidades genéricas. Esto los convierte en servicios de aplicación diseñados para la máxima reutilización. Esto es muy necesario para revisar el inventario existente de servicios de aplicaciones en búsqueda de algún recurso que se esté por diseñar.

A.2.3.2. Confirmar los Contextos.

Cuando se realiza un análisis orientado a servicios es natural confundirse en medio de los requerimientos de negocio. Los servicios de aplicación candidatos producidos en esta fase requieren frecuentemente no tomar los contextos establecidos por medio de servicios de aplicación existentes.

A.2.3.3. Derivar interfaz inicial de servicio.

Utilizando los servicios de aplicación candidatos como la entrada inicial, se debe asegurar que la granularidad de la lógica representada por las operaciones candidatas sea apropiadamente genérica y reutilizable.

Documentar los valores de entrada y salida requeridos para el procesamiento de cada operación candidata y definir estructuras de mensaje utilizando XSD schema.

Completar la abstracción de definiciones de servicio añadiendo los porttype y los message necesarios que contengan los elementos part que referencien tipos de esquema apropiados.

Como resultado de esta fase se obtiene:

Tabla 5. Entradas y salidas de los servicios de aplicación.

| Servicio de Aplicación | Entradas | Salidas |
|---|--|----------------------------|
| Validad Acceso | <ul style="list-style-type: none"> • Usuario • Contraseña • Recurso | Servicios a consumir |
| Consultar factura | <ul style="list-style-type: none"> • Ingreso | Detalles de la facturación |
| Consultar Epicrisis | <ul style="list-style-type: none"> • ID Paciente • ID Ingreso | Epicrisis |
| Administrar Observaciones | <ul style="list-style-type: none"> • ID Factura • ID Observación • Operación • Observación | Observaciones |
| Gestión de acceso valido | <ul style="list-style-type: none"> • Recurso solicitado | Mensaje de Fallo |
| Consultar historia clínica | <ul style="list-style-type: none"> • ID Paciente • ID Ingreso | Historia clínica |
| Consultar imágenes diagnosticas | <ul style="list-style-type: none"> • ID Paciente • ID Ingreso | Imágenes diagnosticas |
| Consultar resultados Laboratorio | <ul style="list-style-type: none"> • ID Paciente • ID Ingreso | Resultado de exámenes |



A.2.3.6. Equipar los servicios candidatos con características especulativas.

- Reusabilidad alta para los servicios de aplicación.
- Agregar Características para el diseño de servicio.
- Esto puede afectar las operaciones existentes, o simplemente agregar nuevas.

A.2.3.7. Identificar las limitaciones técnicas.

Los servicios de negocio así como los servicios de aplicación deben tomar consideraciones de bajo nivel, reales.

- Estudiar el documento de procesamiento de cada operación de servicio más detenidamente.
- Escribir una lista de funciones de procesamiento requeridas para la operación de llevar estos procesamientos.
- Como el procesamiento de la función será ejecutado en un entorno técnico existente.

Establecer acuerdos de nivel de servicio considerando valores como: tiempo de respuesta necesario por servicio, calcular la carga que soportara la arquitectura, el tiempo de operación etc.

Identificar los requerimientos de procesamiento que no pueden ser atendidos por el conjunto de tecnologías de WS.

A.2.4. DISEÑO DE LOS SERVICIOS DE NEGOCIO CENTRADOS EN TAREAS.

La entrada de cada uno de estos procesos de diseño de servicios corresponde con los servicios candidatos producidos en el modelado de servicios durante el análisis orientado a servicios.

A.2.4.1. Definir la lógica del flujo de trabajo.

Contienen lógica de flujo de trabajo embebida que se utiliza para coordinar la composición de servicios en niveles más bajos. Se define la lógica para cada escenario de interacción posible.

El propósito es documentar cada camino de ejecución posible, incluyendo todas las condiciones y excepciones. Los diagramas que resultan de estas actividades se usarán en casos de prueba subsiguientes.

La Figura A.11 expone el flujo de la ejecución de los servicios de aplicación donde por defecto se consulta la epicrisis, en caso de requerir más información el servicio de negocio invocará otros servicios separadamente.

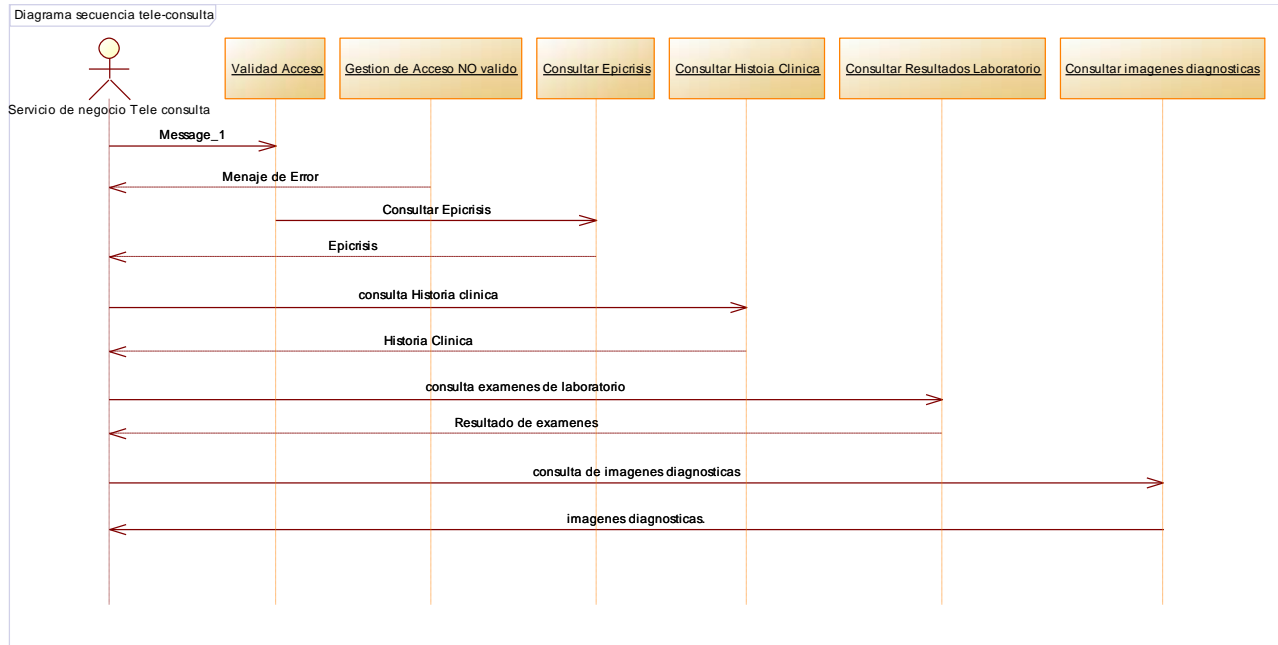


Figura A.11. Diagrama de secuencia servicio tele consulta. flujo comun.

Quando el servicio es invocado con los parámetros necesarios puede retornar otro tipo de respuestas, que incluyan la ejecución y resultados de estas de otros servicios de aplicación. Este caso se expresa en el siguiente diagrama.

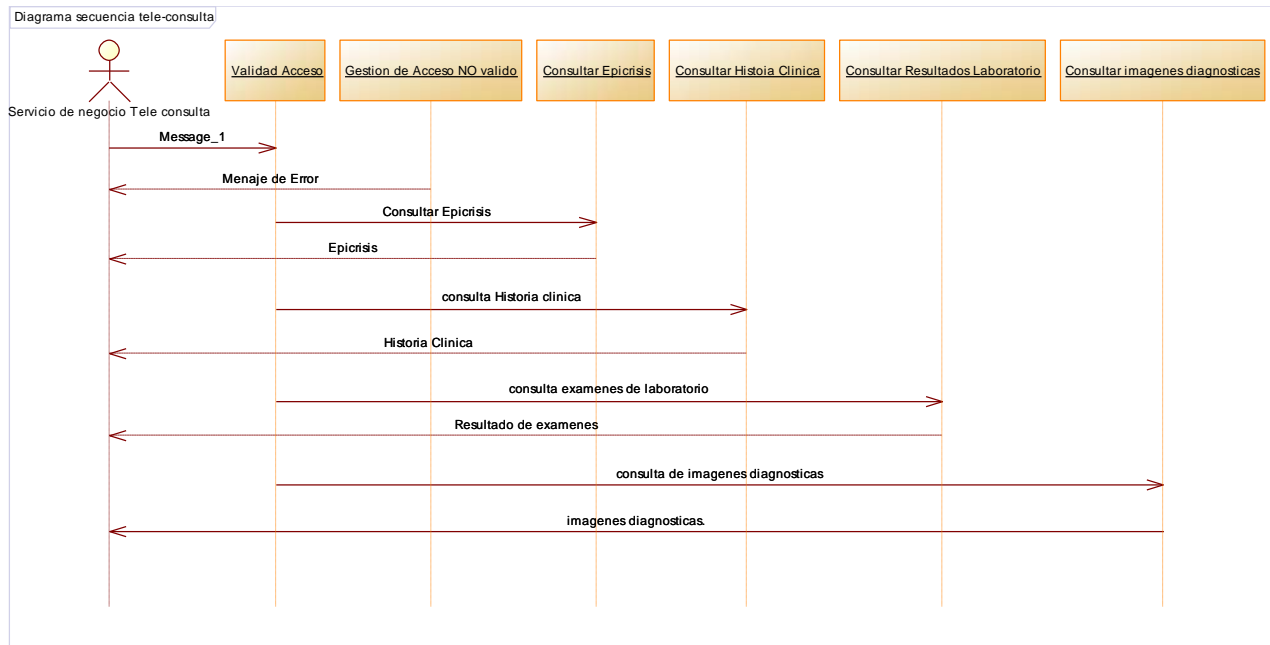


Figura A.12. Diagrama de secuencia servicio de teleconsulta. Parametrizado.

De la misma forma se puede representar el diagrama de secuencia de ambas formas para el proceso de auditoría. se presenta a continuación el diagrama cuando se parametriza la invocación del servicio para que habilite la ejecución de otros servicios.

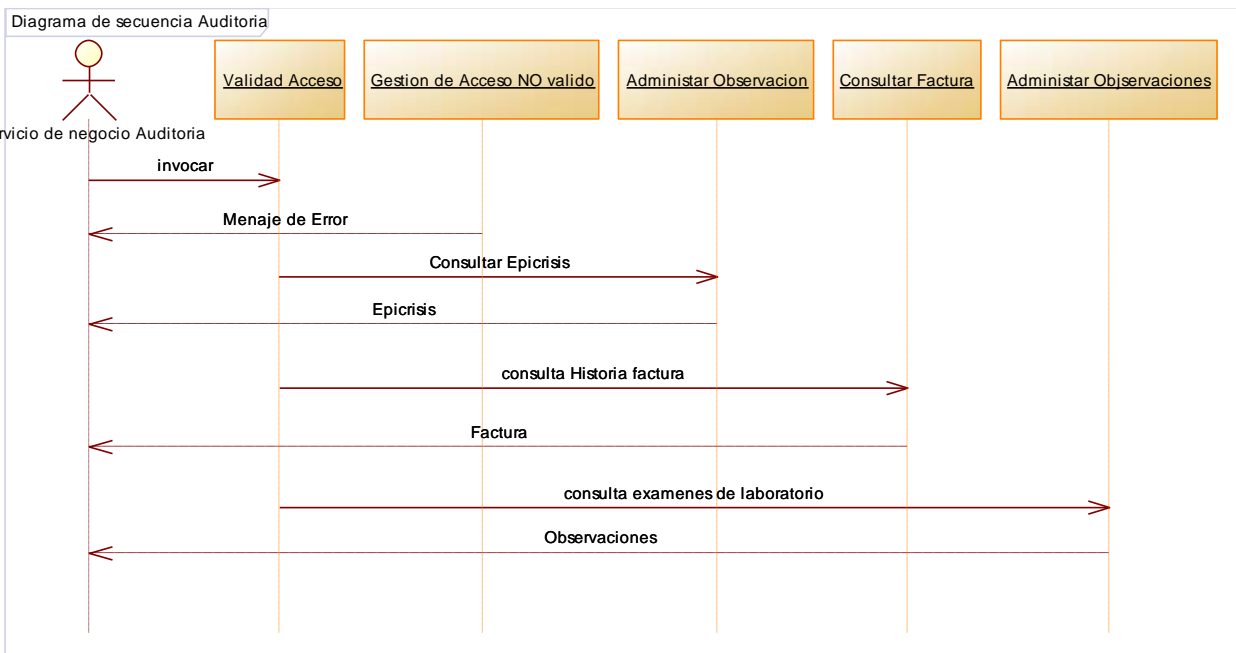


Figura A.13. Diagrama de secuencia de auditoria.

A.2.4.2. Derivar las interfaces de servicio.



En esa fase se recomienda:

- Utilizar las operaciones de servicios de aplicación candidatas para derivar un conjunto de operaciones correspondientes.
- Diferenciar procesos de diseño previo, debido a que, la fuente de donde se derivan las interfaces de servicio en esta fase incluye diagramas de actividad y la lógica del flujo de trabajo ya documentado. Esto permite determinar las tareas adicionales que requiere los servicios centrados en tareas.

Dada la simplicidad de los procesos de negocio se omiten los diagramas de colaboración. No obstante se presenta el diagrama de solicitudes y respuestas de los servicios de negocio.

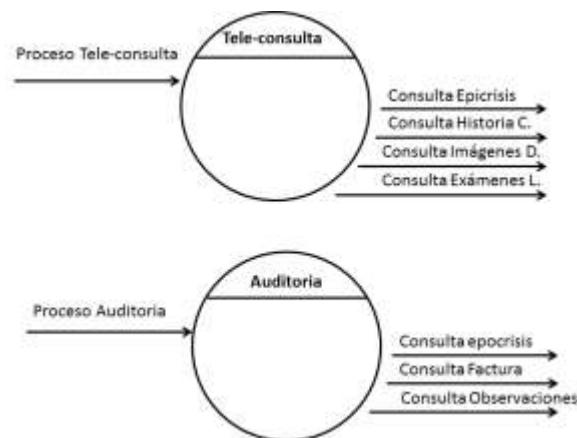
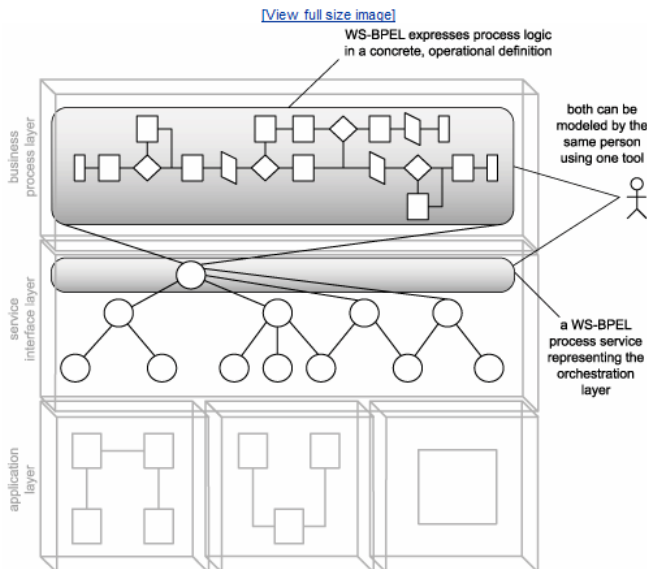


Figura A.14. Solicitudes y respuestas de los servicios de negocio.

A.2.5. DISEÑO DE PROCESOS DE NEGOCIO ORIENTADOS A SERVICIOS.



A.2.5.1. Mapear los escenarios de Interacción.

Definir los requerimientos para el intercambio de mensajes:

- Lógica de flujos de trabajo durante el proceso de modelado de servicios.
- Servicios de proceso candidatos creados.
- Diseño de servicios existentes creados.

Esta información se utiliza para analizar todos los posibles escenarios de interacción entre procesos y partners de servicios. El resultado son los requerimientos de procesamiento que son la base para el diseño de procesos de negocio.

En conclusión en esta etapa definimos las características de los mensajes. Ello se describe en la siguiente tabla, para los servicios de aplicación de los procesos de negocio seleccionados.

Tabla 6. Escenarios de Interacción.

| Servicio de Aplicación | Interacción |
|----------------------------------|--|
| Validad Acceso | Es invocado por cualquier sercicio de negocio. Este servicio reporta el url del servicio a consumir, o el mensaje de error en caso de no validez y el url del servicio gestión de acceso no valido. |
| Consultar factura | Es invocado desde el servicio de negocio Auditoria. Retornan el XML con la información de la factura o el documento HL7, en caso de error retorna vacío. En realidad no importa el orden de la ejecución respecto al servicio consulta de epicrisis |
| Consultar Epicrisis | Retorna epicrisis den XML, o HL7. |
| Administrar Observaciones | Es invocado desde el servicio Auditoria. En los parámetros de acceso al servicio se determina la operación de administración del servicio. En cado de éxito o falla el retorna las observaciones almacenadas. Se pude invocar directamente en cualquier momento. |



| | |
|---|---|
| Gestión de acceso valido | Disparado desde el servicio validar acceso. Retorna Mensaje de Fallo |
| Consultar historia clínica | Historia clínica en los formatos. |
| Consultar imágenes diagnosticas | Retorna los url de los recursos. |
| Consultar resultados Laboratorio | Retorna los resultados de los exámenes en XML. |

A.2.5.2. Diseño de Interfaces de Servicios de Procesos.

Ahora que se entienden los requerimientos de intercambio de mensajes, se puede proceder para definir una definición de servicio para el servicio de proceso. Cuando se trabaja con herramientas de modelado de procesos, el proceso de definición de la WSDL del servicio se generara automáticamente. Se debe contar con la capacidad para editar el código fuente o importar una WSDL propia.

Sugerencias a considerar en esta etapa son:

- Documentar los valores de entrada y salida requeridos para el procesamiento de cada operación.
- La información de metadatos u otros se incluyen en el elemento documentación.

En esta etapa no centraremos el esfuerzo ya que se consideró relevante el uso de un IDE que permita la descripción de servicios.

A.2.5.3. Formalizar la comunicación de servicios partner.

Aquí empieza la definición de procesos BPEL mediante el establecimiento de detalles acerca de los servicios con los cuales nuestros servicios de proceso interactuarán. Se sugiere:

- Definir los servicios partner que estarán participando en el proceso y asignación cada uno de los roles que estarán actuando en un determinado intercambio de mensajes. (Añadir el partnerLinkType al final de la definición de la WSDL para cada servicio)

Esta información documenta esencialmente el flujo de conversación posible que puede ocurrir dentro del curso de la ejecución de proceso. Dependiendo de la herramienta de modelado de procesos utilizado, completando estos pasos puede simplemente requerir interacciones con los proveedores de interfaces de usuario por herramientas de modelado.

A.2.5.4. Definir la lógica de los procesos.

En esta etapa se requiere que toda la inteligencia en los flujos de trabajo existentes sean traspasados e implementados mediante la definición de procesos de negocio vía WS-BPEL.

A.2.5.5. Alinear los escenarios de interacción y refinar los procesos.



Establecer los escenarios de interacción en alineación con la lógica de procesos expresada en la definición de procesos WS-BPEL que proveen un número de beneficios que incluyen:

Los mapas de interacción de servicios son una parte importante de la documentación de la solución y será útil para futuros mantenimientos y requerimientos de transferencia de conocimientos.

La implementación de la lógica de flujos de trabajo originales como un servicio de actividades WS-BPEL puede haber introducido lógica de procesos nuevos o aumentados. La necesidad de interacciones de servicio adicionales requiere descubrir o determinar nuevas condiciones y excepciones que luego puedan ser dirigidas en el proceso de negocio WS-BPEL.

En Figura A.15 y Figura A.16 se representa el diagrama de flujo de proceso de los servicios.

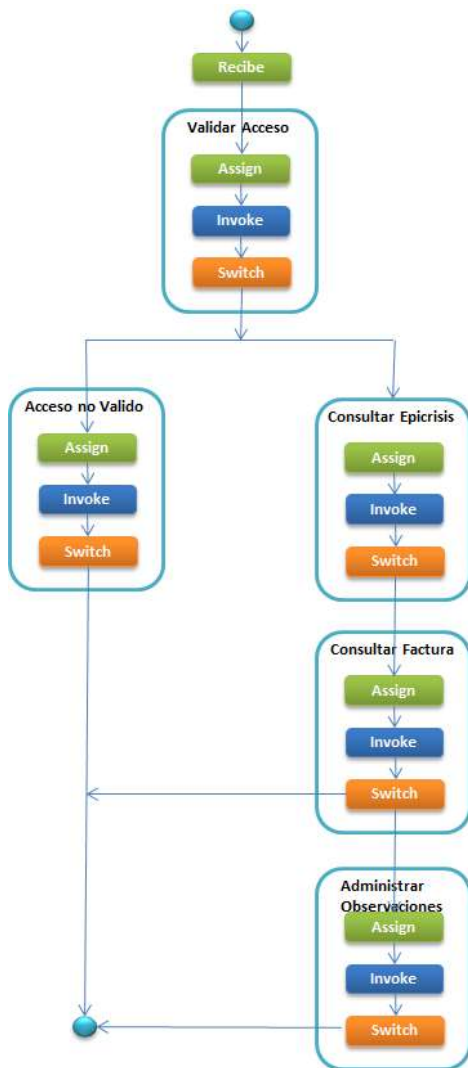


Figura A.15. Proceso Auditoria

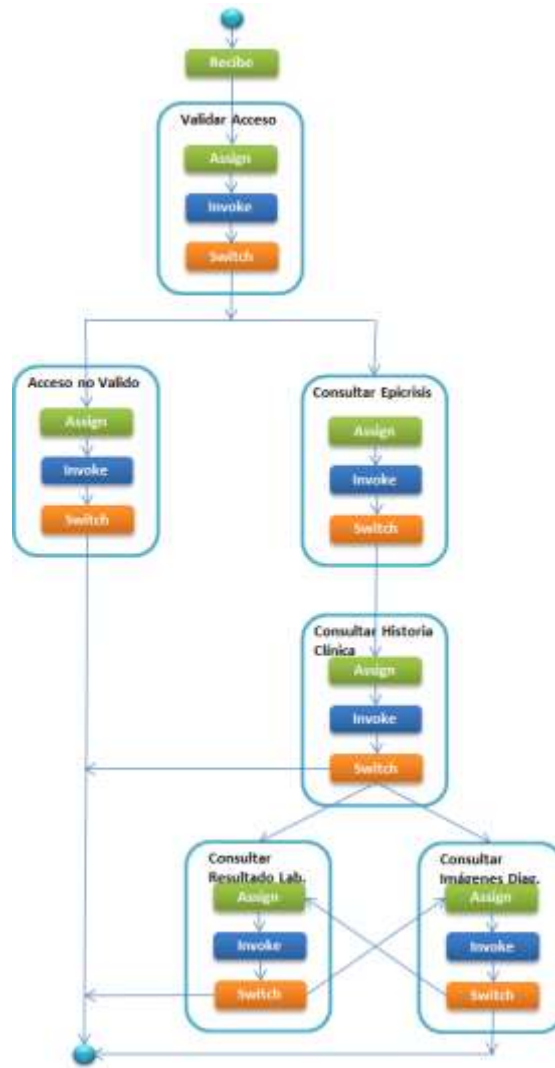


Figura A.16. Proceso Tele-consulta

A.3. DESARROLLO DE SERVICIOS.

La selección de la tecnología es de imperante necesidad, como objetivo principal se seccionan herramientas de libre distribución por lo tanto la solución más adecuada es pensar en JEE, sin embargo es importante un breve espacio de discusión de las más importantes.

JEE y .net son buenas herramientas desarrollo para SOA aquí las ventajas y desventajas, la ventaja de .NET es que se puede desarrollar en distintos lenguajes de programación para la misma plataforma, de esta forma el lenguaje de adopción ya no es barrera. Por su parte JEE tiene un solo lenguaje de programación pero funciona en múltiples plataformas, esto elimina la plataforma como barrera de adopción.



La calidad que tiene JDBC para conectarse a cualquier base de cualquier plataforma le da una ventaja sobre .NET, y la con la Arquitectura de interconexión (JEE Connector Architecture, JCA) provee una solución de interconexión con múltiples servidores de aplicaciones y los EAI.

A.4. PRUEBAS DE SERVICIOS.

Los servicios producto de este diseño de una SOA no requieren niveles de acceso complejos, por lo que se imite el estudio de la implementación de un sistema de nivel de acceso a los recursos del ESB.

Para realizar las pruebas de servicios se emplean las herramientas como OpenScript o SoapUi, para realizar pruebas, selecciona SoapUi por ser sencilla de manejar y por ofrecer parametrizar los datos que van en la solicitudes, de esta forma solo le damos el URL del servicio y configuramos los datos que se quieren enviar, de manera que permite probar excepciones.

Usando SoapUI se probaron los servicios de negocio y de esta forma se terminó que la composición ha sido exitosa.

Nivel de Satisfacción de las políticas de Servicio, son de importancia, para ello se determina un punto de aceptación de los servicios, para esto se aplican pruebas de escalabilidad son el siguiente.

Tabla 7. Pruebas de escalabilidad.

| Numero de Hilos | periodo entre Hilos | Número de veces ejecución |
|------------------------|----------------------------|----------------------------------|
| 60 | 30s | 3 |
| 90 | 20s | 6 |
| 120 | 10s | 9 |

Con esto se estima como nivel de servicio los siguientes valores:

Tabla 8. Niveles de servicio aceptables.

| Propiedad | Valor | Unidad |
|------------------|--------------|----------------------------|
| Latencia | 2134 | Milésimas de segundo |
| Throughout | 600 | Transferencias por segundo |

A.5. DESPLIEGUE DE SERVICIOS.

Distribución de los servicios se muestra en laTabla 9. Distribución de servicios.



Tabla 9. Distribución de servicios.

| Servicio de Aplicación | Lina de negocio | Tipo de servicio |
|---|-----------------------------|------------------------------|
| Validad Acceso | | Control de acceso |
| Consultar factura | facturación | Facturación |
| Consultar Epicrisis | Administración de pacientes | Registro Clínico electrónico |
| Administrar Observaciones | Gestión de calidad | Servicios de auditoría. |
| Gestión de acceso valido | | Control de acceso |
| Consultar historia clínica | Administración de pacientes | Registro Clínico electrónico |
| Consultar imágenes diagnosticas | Imágenes Diagnosticas | Registro Clínico electrónico |
| Consultar resultados Laboratorio | Laboratorio | Registro Clínico electrónico |

Las pruebas de escalabilidad (Anexo J) no ayudaron determinar el impacto de la introducción de elementos el entorno y determinar la escalabilidad del sistema siendo esta óptima.

A.6. ADMINISTRACIÓN DE SERVICIOS.

La administración de servicios en cuanto a crear, eliminar y editar servicios es una tarea que concierne netamente al equipo de administración de sistemas y tecnológica de un empresa, sin embargo, se puede recomendar emplear la composición de servicios para extender las funcionalidades de los existes antes de recurrir la edición, dado el acoplamiento de estos.

Para lograr una administración proactiva se describe determina emplear herramientas de monitoreo estandarizadas, en este caso el protocolo Protocolo Simple de Administración de Red (Simple Network Management Protocol, SNMP) en el anexo I se encuentran guía de su instalación.



B COMPARACIÓN DE ESB DE LIBRE DISTRIBUCIÓN (ANEXO B).

Para la selección de los ESB estudiados, se consideran aquellos desarrollados bajo licencias que permiten su libre distribución.

B.1. OPENESB.

Este ESB hace parte de la comunidad Glassfish², siendo apoyado por un amplio número de miembros, y Sun Microsystems ofrece un soporte comercial, y al estar basado en estándares como JBI³ permite la extensibilidad de componentes (Oracle, 2010).

Tabla B.1. Características de OpenESB.

| Característica | Ok/No | Observación |
|-----------------------------------|--------------|--|
| Enrutamiento | Ok | Emplea JBI Runtime NMR ⁴ Messages. |
| Administración | Ok | Administración WEB de CLI ⁵ (consola de administración). |
| Protocolos transporte | Ok | SOAP, TCP/IP, FTP, HTTP, HTTPS y otros. |
| Transformación de datos | Ok | Empleando el motor de Servicio XSLT ⁶ . |
| Sincronismo y Asincronismo | Ok | Soporta JMS (Java Message Service). |
| Basados en estándares | Ok | Basado en JBI. |
| Seguridad | Ok | Mediante WS-Security, WS-Policy, SAML, SSL entre otros. |
| Definición de Servicios | Ok | Empleando Netbeans. |
| Extensibilidad | Ok | Al estar basado en JBI permite agregar nuevos componentes. |
| Orquestación (BPEL) | Ok | Incluye el motor de servicios BPEL, bajo el lenguaje WS-BPEL. |
| Calidad de Servicio (QoS) | Ok | Mediante JMS y el NMR de JBI, garantiza la entrega de mensajes entre los servicios y/o aplicaciones. |
| Soporte técnico | Ok | Buena documentación, lista de correo, blogs y soporte comercial de Sun Microsystem y Glassfish ESB. |
| IDE | Ok | Empleando Netbeans. |
| Versión estable | Ok | Versión 2.2. Liberada en Diciembre de 2009. |
| Mensajes HL7 | Ok | Emplea HL7bc (HL7 Binding Component). |

B.2. CHAINBUILDER.

Es un ESB desarrollado conforme al estándar JBI, se integra con un extensión para Eclipse, en la Tabla B.2 se describen las características de este producto (ChainBulder Connect, 2010).

² Servidor de aplicaciones de software libre desarrollado por Sun Microsystems, compañía adquirida por Oracle Corporation.

³ JBI, de las siglas en ingles: Java Business Integration, es una especificación desarrollada bajo la Java Community.

⁴ NMR, de las siglas en ingles: Network Message Routing.

⁵ CLI, de las siglas Client Line Interface.

⁶ XSLT, de las siglas en ingles: Extensible Stylesheet Language Transformations especificación para la transformación de formatos.



Tabla B.2. Características de ChainBuilder.

| Característica | SI/NO | Observación |
|-----------------------------------|-----------|---|
| Enrutamiento | Si | Se brinda por medio del Conted-Based Router (CBR), el cual identifica los mensajes y los direcciona según reglas definidas. |
| Administración | Si | Administración Web basada empleando AJAX. |
| Protocolos de transporte | Si | HTTP, FTP, HTTPS, SOAP, TCP/IP POP3, SMTP y otros. |
| Transformación de datos | Si | Con el estándar XSLT. |
| Sincronismo y Asincronismo | Si | Patrones de mensajería de solicitud y repuesta Asíncrona. |
| Basados en estándares | Si | Implementa la especificación JBI, adicionalmente soporta XML, JMS, WSDL, SOAP, entre otros. |
| Seguridad | Si | Usa de SSL, WS-Security, SAML y WS-Trust para mensajes. |
| Definición de Servicios | No | No cuenta con herramienta de definición de servicios. |
| Extensibilidad | Si | Al estar basado en JBI. |
| Orquestación (BPEL) | Si | Empleando BPEL. |
| Calidad de Servicio (QoS) | Si | Entrega segura de mensajes con la implementación de JMS ActiveMQ. |
| Soporte técnico | Si | Documentación poco específica, lista de correo y foros. Provee soporte comercial al adquirir la versión Profesional. |
| IDE | Si | GUI basado en una extensión de Eclipse para la edición de flujos, transformación de datos, creación de componentes JBI |
| Versión estable | Si | 3.0 M2 liberada del 7 de agosto de 2009 |
| Mensajes HL7 | Si | Usando el parzer HL7. |

B.3. FUSE ESB.

Derivando de Apache ServiceMix, también implementado bajo la especificación JBI y OSGi⁷ (FUSE Community, 2010), se integra a herramientas para el desarrollo de una SOA: FUSE Message Broker (una implementación de JMS para el intercambio de mensajes), FUSE Mediation Router (herramienta para integración de servicios, aplicaciones y protocolos de transporte), FUSE Services Framework (plataforma para crear Servicios Web). FUSE ESB es una robusta infraestructura para la integración de aplicaciones que permite la interacción y despliegue de los servicios desarrollados con FUSE Services Framework.

Tabla B.3. Características de FUSE ESB.

| Característica | SI/NO | Observación |
|-----------------------------------|-----------|--|
| Enrutamiento | Si | FUSE Mediation Router el cual está basado en Apache Camel. |
| Administración | Si | Web, muy limitada en funcionalidades. |
| Protocolos de transporte | Si | HTTP, HTTPS, JMS, FTP, XMPP, SOAP, entre otros. |
| Transformación de datos | Si | Soporte para XSLT (Saxon) y XQuery. |
| Sincronismo y Asincronismo | Si | Implementación de JMS. |
| Basados en estándares | Si | Soporte para JBI, JMS, JCA y JMX. |
| Seguridad | Si | Adaptado con FUSE Message Broker, provee transporte de mensajes seguro mediante HTTPS y SSL. |
| Definición de Servicios | Si | Implementando en FUSE Services Framework. |

⁷ Es un Framework de módulos y servicios en Java.



| | | |
|----------------------------------|-----------|---|
| Extensibilidad | Si | Al implementar JBI. |
| Orquestación (BPEL) | Si | Provee una extensión estándar que soporta cualquier motor BPEL basado en JBI. |
| Calidad de Servicio (QoS) | Si | Entrega segura de mensajes mediante JMS. |
| Soporte técnico | Si | Documentación, foros, listas de correo y soporte comercial. |
| IDE | Si | Mediante FUSE IDE (extensión para eclipse) |
| Versión estable | Si | Versión 3.4.0.4 liberada el 31 de Julio del 2009. |
| Mensajes HL7 | Si | Integrándolo con Apache Camel ⁸ HL7. |

B.4. PETALS ESB.

PEtALS es un ESB de fuente abierta de OW2 middleware consortium (Petals ESB, 2010) basado en la licencia LGPL. Cuenta con un motor de enrutamiento que consiste de tres módulos: resolución de direcciones, control de acceso y transporte. El diseño de la mediación y direccionamiento está basado en patrones para la integración empresarial EIP. Soporta estándares comunes como WSDL, SOAP, REST, POP, SMTP, IMAP, EJB y JDBC. Existe una gran comunidad de soporte así como documentación comprensible.

Tabla B.4. Características de Petals ESB.

| Característica | SI/NO | Observación |
|-----------------------------------|--------------|---|
| Enrutamiento | Si | Mediación y Enrutamiento basado en Patrones de Integración Empresarial-EIP. |
| Administración | Si | Interfaz de administración Web. |
| Protocolos de Transporte | Si | HTTP, SOAP, REST, POP, SMTP, IMAP, FTP. |
| Transformación de datos | Si | Soporta XSLT. |
| Sincronismo y Asincronismo | Si | Implementación de JMS. |
| Basados en estándares | Si | Soporte para JBI y EJB. |
| Seguridad | Si | Soporte para HTTPS. |
| Definición de Servicios | Si | Soporta EIP, POJO y SCA |
| Extensibilidad | Si | Soporte bajo JBI. |
| Orquestación (BPEL) | Si | Soporta BPEL. |
| Calidad de Servicio (QoS) | No | La documentación no especifica esta opción. |
| Soporte técnico | Si | Documentación, foros, listas de correo y soporte comercial. |
| IDE | Si | Petals Studio (extensión para Eclipse). |
| Versión estable | Si | Versión 3.0.3 liberada el 15 de Enero de 2010. |
| Mensajes HL7 | No | No soporta HL7. |

B.5. JBossESB.

Es parte de una SOI (Infraestructura Orientada a Servicios). Soporta gran variedad de protocolos como HTTP, FTP, entre otros. También provee herramientas visuales para el diseño de flujos de procesos con BPEL, un contenedor de aplicaciones, un repositorio de metadatos, entre otros (JBoss, 2010).

⁸ Librería para trabajar con el protocolo HL7 del proyecto Camel.



Tabla B.5. Características de JBoss ESB.

| Característica | SI/NO | Observación |
|-----------------------------------|-----------|---|
| Enrutamiento | Si | Soporta enrutamiento basado en Contenido utilizando JBossRules. JBossMQ, JBoss Messaging, Oracle AQ and MQSeries. |
| Administración | Si | Editor de configuración web. |
| Protocolos de transporte | Si | SOAP, HTTP, SMTP. |
| Transformación de datos | Si | Soporta XSLT y Smook. |
| Sincronismo y Asincronismo | Si | Soporta implementaciones de JMS. |
| Basados en estándares | Si | EJB, SCA, JMS, POJO, JCA. |
| Seguridad | Si | Soporta HTTPS y SFTP. También WS-Security; el componente SecurityService permite manejar la seguridad de servicios. |
| Definición de Servicios | Si | Soporte para POJO y SCA. |
| Extensibilidad | Si | Al soportar JBI |
| Orquestación (BPEL) | Si | Soporta BPEL. |
| Calidad de Servicio (QoS) | Si | La entrega de mensajes es segura ya que soporta JMS. |
| Soporte técnico | No | Documentación escasa, como motivación a adquirir un soporte bajo licencia. |
| IDE | Si | JBoss Tools (extensión para Eclipse). |
| Versión estable | Si | JBoss ESB 4.9 liberada el 10 de Agosto de 2010. |
| Mensajes HL7 | Si | Integración con Apache Camel. |

B.6. APACHE SYNAPSE.

Apache Synapse ESB está diseñando para ser simple y liviano con el objetivo de ofrecer alto rendimiento. Creado sobre una base totalmente asíncrona, con soporte para XML, servicios web, formatos binarios y texto. La configuración de Apache Synapse consta esencialmente de dos archivos XML: el archivo de configuración (synapse.xml), donde se configuran las reglas de mediación y configuración del ESB, y el archivo de configuración del motor Axis2 (axis2.xml), el cual configura el soporte para Servicios Web (Apache Software Foundation, 2008).

Tabla B.6. Características de Apache Synapse.

| Característica | SI/NO | Observación |
|-----------------------------------|-----------|---|
| Enrutamiento | Si | Basado en Regex. |
| Administración | Si | Mediante archivos XML. |
| Protocolos de transporte | Si | HTTP, FTP, SMTP, POP 3, IMAP, SMTP. |
| Transformación de datos | Si | XSLT, XPath, XQuery. |
| Sincronismo y Asincronismo | Si | Soporte para JMS. |
| Basados en estándares | Si | Soporta JMS XML and SOAP/JMS |
| Seguridad | Si | WS-Security y WS-Reliable Messaging. |
| Definición de Servicios | No | No soporta mecanismos de definición de servicios. |
| Extensibilidad | No | De bajo nivel con Java, Ruby o Groovy. |
| Orquestación (BPEL) | No | No existen mecanismos de orquestación. |
| Calidad de Servicio (QoS) | Si | Entrega segura de mensajes mediante JMS. |
| Soporte técnico | No | Documentos, ejemplos y una guía práctica. |
| IDE | No | No se especifica en la documentación una herramienta. |



| | | |
|------------------------|-----------|--|
| Versión estable | Si | Versión 1.2 Liberada el 9 de Junio de 2008. |
| Mensajes HL7 | No | Se debe adaptar librerías de otros proyectos para habilitarlo. |

B.7. APACHE SERVICE MIX.

Este ESB basado en el estándar Java Business Integration (JBI), combina la funcionalidad de una SOA y una Arquitectura Manejada por Eventos (EDA). Está integrado al servidor Apache Gerónimo, lo que le permite desplegar servicios y componentes JBI. Incorpora una extensión de Spring para facilitar entre otras cosas, el desarrollo de componentes JBI. ServiceMix forma parte del “Apache Software Foundation” (Apache Software Foundation, 2010).

Tabla B.7. Características de ServiceMix.

| Característica | SI/NO | Observación |
|-----------------------------------|-----------|--|
| Enrutamiento | Si | Normalized Message Router (NMR) |
| Administración | Si | Interface Web. |
| Protocolos de Transporte | Si | HTTP, JMS, FTP, SOAP. |
| Transformación de datos | Si | Service Engine Saxon, que utiliza XSLT. |
| Sincronismo y Asincronismo | Si | JMS con ActiveMQ. |
| Basados en estándares | Si | JBI y XML. |
| Seguridad | Si | WS-Security. Java Authentication and Authorization Service (JAAS). |
| Definición de Servicios | No | No se especifica la definición de servicios. |
| Extensibilidad | Si | Soporta JBI. |
| Orquestación (BPEL) | No | Sin embargo, puede integrarse un motor de BPEL. |
| Calidad de Servicio (QoS) | Si | Implementa JMS, por lo tanto, la entrega de mensajes es segura. |
| Soporte técnico | Si | Soporte al proyecto Open Source, así como al comercial. Comunidad con documentación, foros y manuales de implementación. |
| IDE | Si | Extensión de Eclipse. |
| Versión estable | Si | Versión 4.2.0. |
| Mensajes HL7 | Si | Complemento servicemix-hl7 (es un componente JBI). |

B.8. WSO2 ESB.

Basado en Apache Synapse, tiene un soporte para XML, Servicios Web, conexiones y mensajes concurrentes. Además, provee una GUI para configurar, manejar, monitorear y ejecutar el servidor ESB que consta de un editor de secuencias, monitor de registros, entre otras utilidades. Su manejo requiere de los conocimientos de Apache Synapse. WSO2 puede manejar balanceo de carga, regular el flujo y la caché en entornos con almacenamiento en clúster para proveer alta disponibilidad y escalabilidad (WSO2, 2010).

Tabla B.8. Características de WSO2 ESB.

| Característica | SI/NO | Observación |
|---------------------|-----------|----------------------------------|
| Enrutamiento | Si | Normalized Message Router (NMR). |



| | | |
|-----------------------------------|-----------|---|
| Administración | Si | Interfaz gráfica. |
| Protocolos de transporte | Si | HTTP, HTTPS, FTP, FTP, SOAP, SMTP. |
| Transformación de datos | Si | Utiliza XSLT. |
| Sincronismo y Asincronismo | Si | Con soporte para JMS. |
| Basados en estándares | Si | XML, WS, SOAP. |
| Seguridad | Si | HTTPS. |
| Definición de Servicios | No | No cuenta con herramienta para la definición de servicios. |
| Extensibilidad | Si | Dado que soporta JBI. |
| Orquestación (BPEL) | NO | Se puede integrar un motor como de orquestación. |
| Calidad de Servicio (QoS) | Si | Entrega segura de mensajes mediante JMS. |
| Soporte técnico | Si | Comunidad del proyecto open source, listas de correo, foros y manuales de implementación. |
| IDE | No | No se especifica un entorno de desarrollo determinado |
| Versión estable | Si | Versión 3.0.1 libreada el 8 de Septiembre de 2010. |
| Mensajes HL7 | No | No se especifica un componente para soportarlo. |

B.9. MULE ESB.

MULE es un framework de mensajería hecho en JAVA que permite el intercambio de mensajes y la comunicación entre aplicaciones. Este ESB hace un manejo transparente de todas las iteraciones entre las aplicaciones y componentes, independientemente del protocolo de transporte usado; diferencia de los otros ESB, los cuales necesitan de un adaptador o especificar el protocolo para conectar las aplicaciones al Bus y además necesitan convertir los datos en un formato de mensajes único, en MULE los mensajes son enviados a través de un canal de comunicación como HTTP o JMS y son traducidos solo de ser necesario (Mulesoft, 2010).

Tabla B.9. Características de Mule ESB.

| Característica | SI/NO | Observación |
|-----------------------------------|--------------|---|
| Enrutamiento | Si | Cuatro tipos de enrutamiento: inbound, outbound, Asynchronous Reply y Cath all. |
| Administración | No | No dispone de una GUI de administración. |
| Protocolos de transporte | Si | HTTP, HTTPS, FTP, SOAP, POP3, SMTP |
| Transformación de datos | Si | Soporte para XSLT. |
| Sincronismo y Asincronismo | Si | Implementa JMS y MEP. |
| Basados en estándares | Si | XML. |
| Seguridad | Si | SSL y HTTPS. |
| Definición de Servicios | No | No ofrece una herramienta para la definición de servicios. |
| Extensibilidad | No | De baja extensibilidad, solo si se crean componente son los frameworks: spring, hivemind y plexus |
| Orquestación (BPEL) | Si | Soporte para BPEL. |
| Calidad de Servicio (QoS) | Si | Soporte de JMS. |
| Soporte técnico | Si | Soporte gratuito (Comunidad Mule), existe documentación y foros, artículos y blogs. Ofrecen soporte licenciado (Mule Enterprise). |
| IDE | Si | Mule IDE. Además existe una extensión para Eclipse. |
| Versión estable | Si | Versión 3.0.0 librada el 21 de Mayo de 2010 |
| Mensajes HL7 | Si | Con componentes del proyecto Mirth Mith (interface de mensajes HL7). |



B.10. JEEESB.

Es una nueva iniciativa que está en desarrollo, sus objetivos es crear un ESB con las principales funciones de enrutamiento de mensajes. Actúa como un mediador en una aplicación Java. La documentación para este ESB es casi nula (Jee(G)ESB, 2010).

Tabla B.10. Características JeeESB.

| Característica | SI/NO | Observación |
|-----------------------------------|-----------|--|
| Enrutamiento | Si | Basado en reglas de configuración. |
| Administración | No | Aplicación web que edita los archivos XML de configuración. |
| Protocolos | Si | HTTP y SOAP. |
| Transformación de datos | Si | XPath y Xquer. |
| Sincronismo y Asincronismo | No | No hay información . |
| Basados en estándares | Si | XML y SOAP. |
| Seguridad | No | No hay información. |
| Definición de Servicios | No | No hay información. |
| Extensibilidad | No | No hay información. |
| Orquestación (BPEL) | No | La documentación dice que es un objetivo, pero no especifica el motor de orquestación. |
| Calidad de Servicio (QoS) | No | No hay información. |
| Soporte técnico | No | El proyecto no ha alcanzado la suficiente madures como para ofrecer documentación. |
| IDE | No | No se especifica. |
| Versión estable | No | No hay lanzamiento estable, solo los SVN para desarrollo. |
| Mensajes HL7 | No | No hay información. |

B.11. CELTIX.

Celtix está basado en la especificación JAX-WS 2.0, logrando una completa compatibilidad con los Servicios Web. Soporta HTTP y SOAP como protocolos de transporte y cualquier implementación de JMS (Java Message Service) para el intercambio de mensajes. Además provee un conjunto de API que permiten adicionar funcionalidades (ObjectWeb, 2008).

Tabla B.11. Características de Celtix.

| Característica | SI/NO | Observación |
|-----------------------------------|-----------|--|
| Enrutamiento | Si | Normalized Message Router (NMR). |
| Administración | No | Solo mediante línea de comandos. |
| Protocolos de transporte | Si | HTTP y SOAP. |
| Transformación de datos | Si | Emplea XSLT. |
| Sincronismo y Asincronismo | Si | Implementa JMS. |
| Basados en estándares | Si | JBI, WSDL, SOAP. |
| Seguridad | Si | HTTPS y WS-ReliableMessaging. |
| Definición de Servicios | No | No disponible. |
| Extensibilidad | Si | La que ofrece el desarrollo bajo el estándar JBI |
| Orquestación (BPEL) | Si | Empleando BPEL. |
| Calidad de Servicio (QoS) | No | No se especifica que lo soporte. |



| | | |
|------------------------|-----------|--|
| Soporte técnico | Si | Comunidad de proyecto open source, foros y manuales de implementación. |
| IDE | SI | Extensión de Eclipse. |
| Versión estable | SI | Versión 1.0 liberada el 8 de Mayo del 2006. |
| Mensajes HL7 | Si | Mediante JEngine's JMX incluye un servidor y cliente HL7 2.x. |

B.12. RESUMEN DE RESULTADOS

La siguiente tabla resume los resultados y permite hacer una comparación rápida de las diferentes alternativas estudiadas.

Tabla B.12. Comparación de los ESB.

| Herramientas Consideradas | Criterios de Selección | | | | | | | | | | | | | | |
|---------------------------|------------------------|-----------------|---------------------------|-----------------|--------------------------------------|-----------------------|------------|--------------------------|-----------------|----------------------------|----------------------|------------------|-----------------------------------|------------------|-------------------|
| | Enrutamiento | Administración. | Protocolos de Transporte. | Transformación. | Mensajes Sincrónicos y Asincrónicos. | Basado en Estándares. | Seguridad. | Definición de Servicios. | Extensibilidad. | Orquestación de Servicios. | Calidad de Servicio. | Soporte Técnico. | Entorno de Desarrollo. Integrado. | Versión Estable. | Soporte para HL7. |
| OpenESB | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ChainBuilder ESB | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Fuse ESB | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| PEtALS | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ |
| JBoss ESB | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ |
| Apache Synapse | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ |
| Apache ServiceMix | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |
| WS02 ESB | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ |
| Mule ESB | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| JeeESB | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Celtix | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |



C INSTALACIÓN DE OPEN ESB

C.1. DESCARGAR OPENESB 2.2

Desde la página de OpenESB, sección Download, se accede a los enlaces de descargas de GlassFish ESB v2.2. Se ofrecen dos distribuciones:

- Full Install (195 MB), versión con el bundle GlassFish 2.2 y NetBeans 6.7.1 con los módulos y componentes configurados e instalados tanto en el servidor como en el IDE para propósitos SOA. Asimismo, la integración entre GlassFish y NetBeans está perfectamente resuelta (este es el descargado).
- Sólo componentes (66 MB). Esta distribución parte de tener instaladas en nuestro equipo las versiones estándar de NetBeans 6.7.1 y GlassFish 2.2. Estos componentes (Java Business Integration (JBI), Binding Component, Service Engines, SOA Project Modules) se instalarán en ambas plataformas, enriqueciéndolas con capacidades de tipo SOA.

C.2. INSTALACIÓN DE OPENESB.

Se describirá la secuencia de pasos necesaria para comprender y hacer la instalación de OpenESB.

Se debe ejecutar el instalador `glassfishesb-v2.2-full-installer-windows.exe` y se siguiendo los pasos del asistente:



Figura C.1 Asistente de instalación OpenESB

Si se pulsa el botón Customize se verá claramente los componentes a instalar: el GlassFish ESB DesignTime, que corresponde con NetBeans 6.7.1 con plugins SOA, y GlassFish ESB Runtime, que es el servidor GlassFish con JBI integrado y configurado.

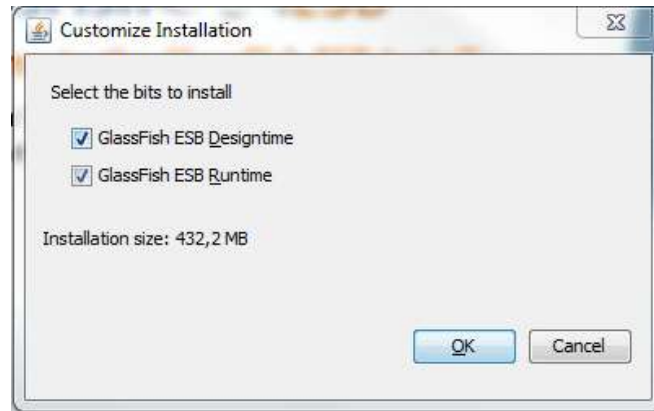


Figura C.2 Personalizar Instalación

En el siguiente paso se acepta el acuerdo de licencia:

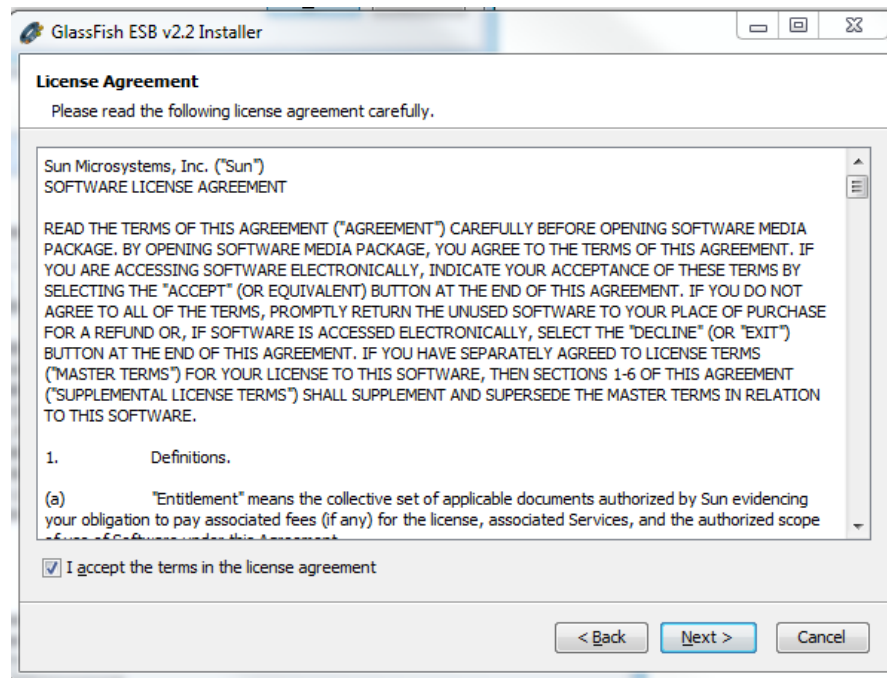


Figura C.3 Terminos de licenciamiento

A continuación se configura el entorno del IDE Netbeans:

- En el área de texto superior, incluir la ruta absoluta donde se instalará NetBeans. La ruta por defecto, C:\GlassFishESBv22\netbeans, es adecuada y la aceptamos (en todo caso, al instalarlo en Windows Vista con UAC activado hay que tener precaución de no indicar directorios protegidos por el sistema operativo: C:\, C:\Windows, C:\Users, C:\Program Files)
- En el campo inferior deberá introducirse la ruta de nuestra instalación de Java en el equipo. Normalmente este campo estará completado, tomando el valor de la variable del



sistema JAVA_HOME definida en la configuración de la JDK, que incluye las librerías y binarios de compilación y la JRE.

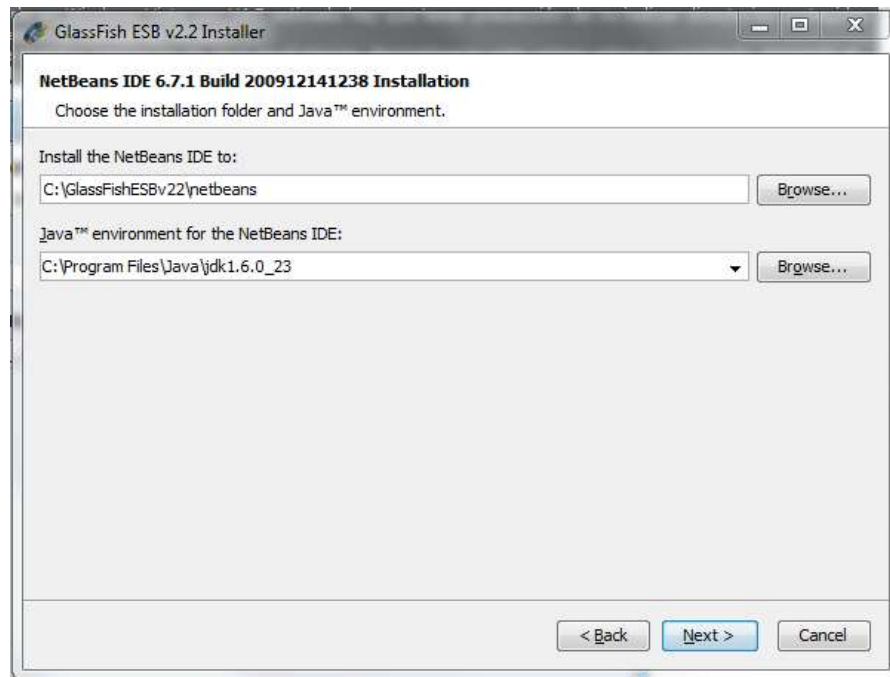


Figura C.4 Configuración de entorno Java.

Se llega el turno de configurar GlassFish 2.2:

- El primer campo permite introducir la ruta de instalación del servidor. Al igual que en el paso anterior con la de NetBeans, se puede aceptar dicha ruta.
- El segundo campo requiere la ruta de instalación de la JDK, su valor corresponderá normalmente con JAVA_HOME
- Admin Username: usuario administrador para la consola de administración de GlassFish.
- Admin Password/Retype Password: contraseña para el usuario administrador anterior. el valor por defecto: adminadmin
- A continuación configurar los puertos para las conexiones con el servidor. Resulta útil para evitar conflictos con otros servidores, aplicaciones instaladas o bien otras configuraciones (firewalls, etc). Es conveniente conocer el convenio de uso de puertos TCP y UDP de servidores basados en Apache HTTP Server u otros como Jetty, etc. (GlassFish está basado en el framework Grizzly, según me explica Raúl Expósito). En este caso, serán:
 - HTTP Port: 8080. Conexiones HTTP
 - HTTPS Port: 8181. Conexiones HTTP seguras
 - Admin Port: 4848. Puerto de la consola de administración de GlassFish
 - JMS Port: 7676. Puerto de la implementación del API del Servicio de Mensajería de Java para el uso de colas basada en Java EE
 - JMX Admin Port: 8686. Puerto para las Java Management Extensions, tecnología para gestionar y monitorizar diversos tipos de aplicaciones



- IIOOP Port: 3100. Puerto para el protocolo IIOOP (Internet Inter-ORB Protocol) del OMG, implementado en CORBA. Su propósito es la integración basada en la comunicación de objetos independientemente del lenguaje de programación que los implemente.
- IIOOP SSL Port: 3820. Comunicación segura para el protocolo IIOOP
- IIOOP Mutual Auth Port: 3920. Puerto para la autenticación de doble sentido entre dos actores de una comunicación, sobre IIOOP en este caso.

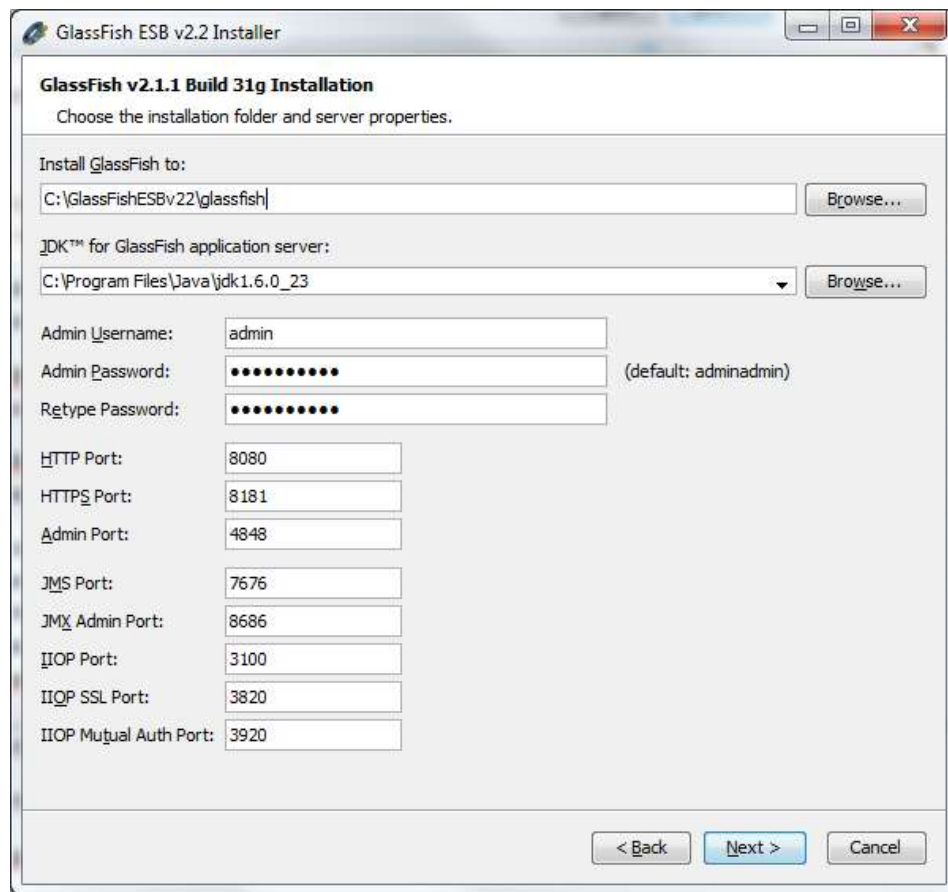


Figura C.5. Configuración de servicios.

En el siguiente paso se muestra el resumen de la configuración de instalación. Se procede con la instalación.

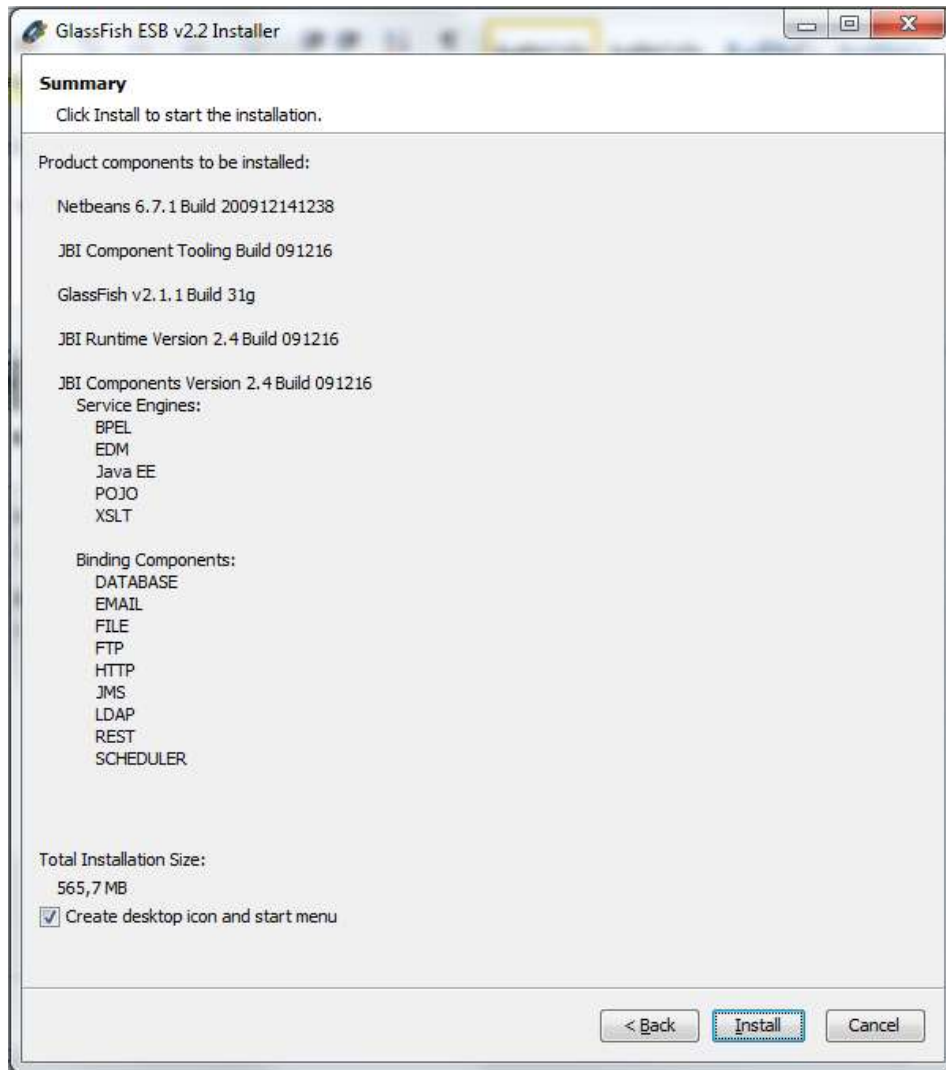


Figura C.6. Resumen de configuración.

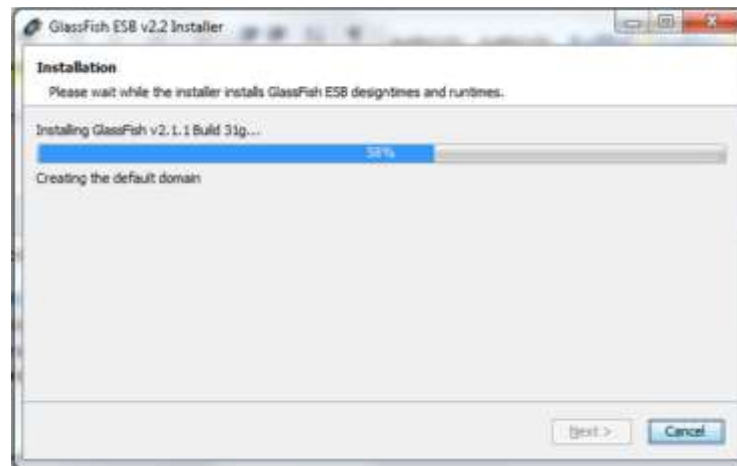


Figura C.7. Instalación de OpenESB.



Tras unos minutos, el proceso habrá finalizado y nos ofrecen el registro como usuario de productos de Sun.

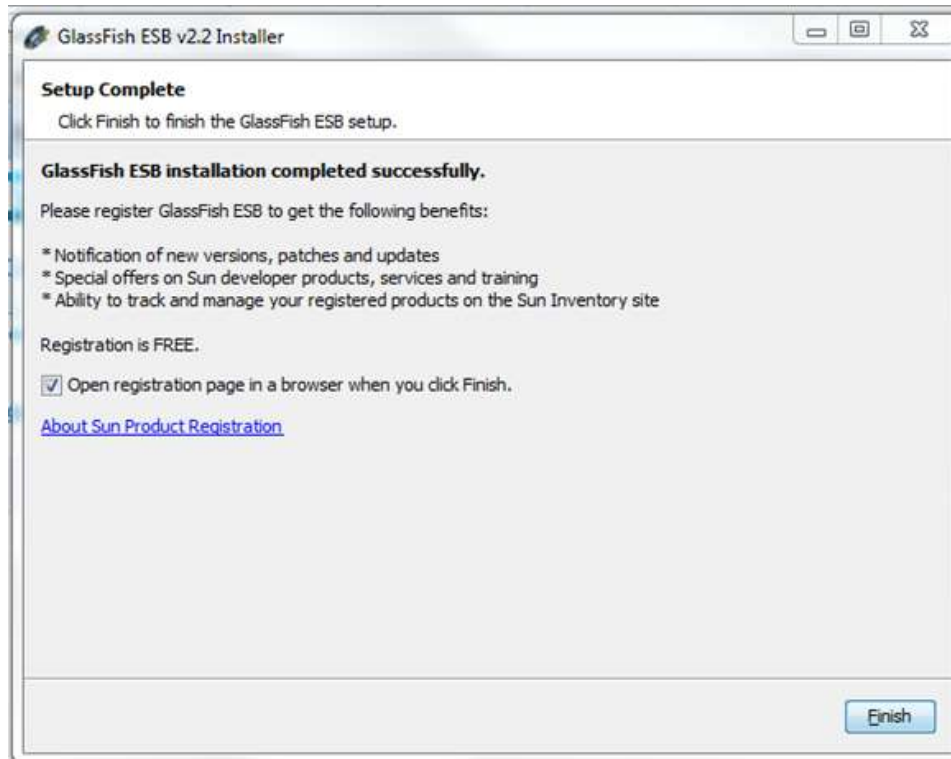


Figura C.8. Instalación completa openESB.

C.3. INTRODUCCIÓN A GLASSFISH ESB 2.2

La implementación de un modelo SOA utilizando las capacidades de un ESB con OpenESB se realiza a través de NetBeans, que nos permite el diseño, desarrollo, despliegue y control de los servicios e infraestructura del Enterprise Service Bus.

C.3.1. IDE NETBEANS CON CAPACIDADES SOA

NetBeans es un IDE cuya funcionalidad puede ampliarse mediante plugins. La distribución incluida con OpenESB incluye un conjunto inicial de ellos que nos permitirá implementar conceptos de una SOA.

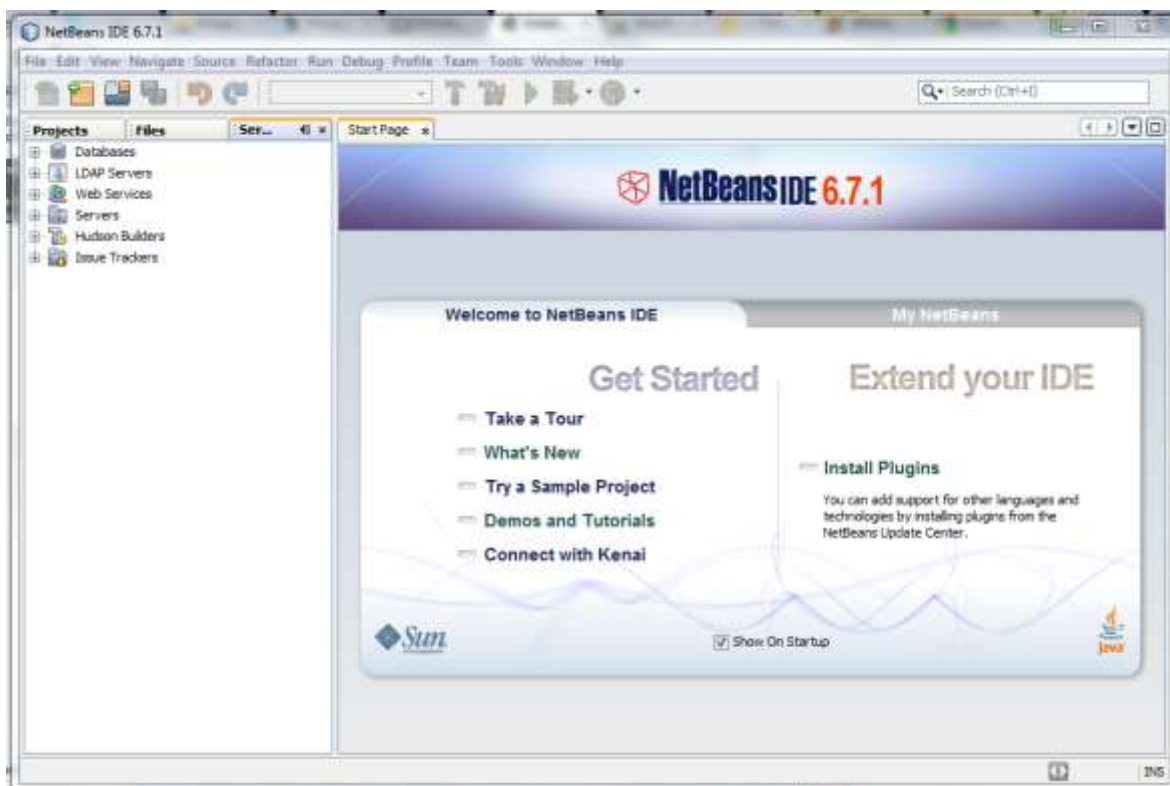


Figura C.9. Netbeans IDE.

Al crear un nuevo proyecto dispondremos ahora de la categoría SOA.

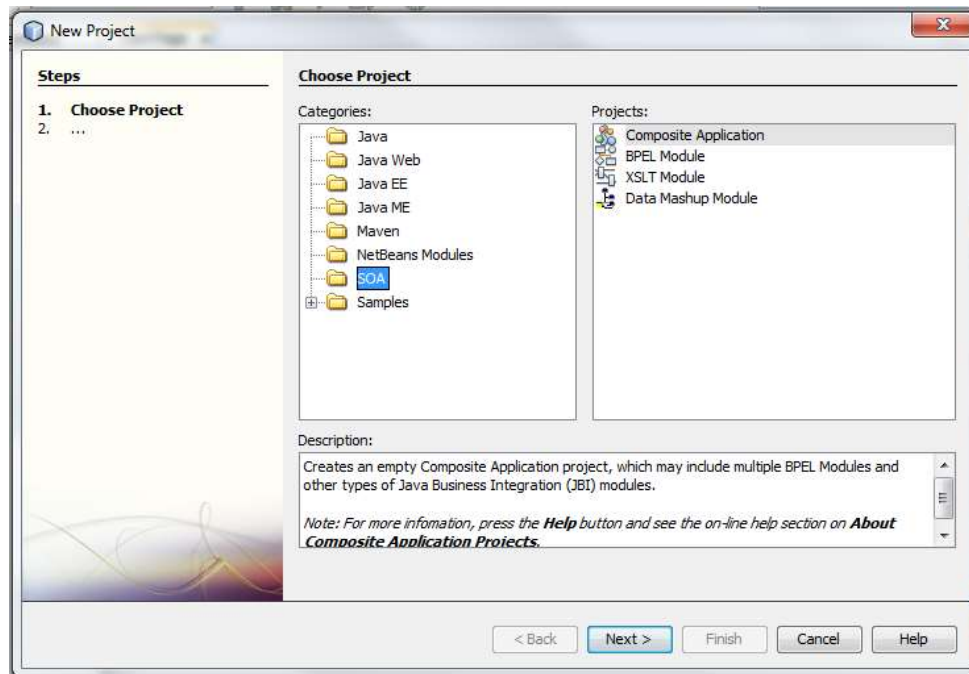


Figura C.10. Capacidades para SOA Netbeans IDE.

Los tipos de proyectos que podremos diseñar e implementar son:

- **Composite Application.** En un proyecto de este tipo simplemente se añaden otros proyectos SOA y de servicios web. Cada proyecto incluido se denomina componente o módulo, y son desplegados juntos en el contenedor JBI. Su interacción en runtime implementará una funcionalidad de negocio de SOA.
- **BPEL Module.** Los procesos de negocio que involucran varios servicios web u otros recursos pueden modelarse gráficamente y luego ejecutarse a través de un módulo BPEL. Su capacidad y propósito son mucho más limitados que un BPM, su finalidad es la orquestación de servicios para realizar composición de éstos, práctica esencial en SOA.
- **XSLT Module.** Una de las capacidades de un ESB es la transformación de los mensajes XML que se enrutan a través del Bus. Las transformaciones de un tipo (los DTD, Schema definen 'tipos') de XML a otro se realizan mediante XSLT, y este módulo nos permite hacerlo.
- **Data Mashup Module.** Consiste en combinar varias fuentes de datos (hojas de cálculo, ficheros, bases de datos, servicios, etc) y presentarlo al servicio consumidor en un formato normalizado, como si fuese un único recurso (e.g., como tablas de base de datos). El ESB hace las tareas de composición de los datos y gestiona sus orígenes, de manera transparente al consumidor que únicamente se centra en su lógica sobre ellos.
- **Intelligent Event Processing Module.** SOA tiene características de Arquitecturas Orientadas a Eventos, de manera que éstos pueden ser capturados y combinados, agregados, filtrados, almacenados, etc, según nuestros intereses de negocio. La información procesada a partir de los eventos es esencial para las arquitecturas SOA más avanzadas (nivel de madurez).

C.3.2. SERVIDOR GLASSFISH CON FUNCIONALIDAD ESB



SOA se basa, en esencia, en que los servicios interoperen entre sí. Para ello hace falta una infraestructura que administre esa comunicación: el ESB. OpenESB es una suma de funcionalidades desplegadas en el servidor GlassFish, que en su conjunto lo convierte en un ESB.

Tras la instalación disponemos de un servidor GlassFish con sus funcionalidades habituales (Aplicaciones, Recursos), al que se añade un contenedor JBI en el que se despliegan componentes funcionales propios de un ESB, además de nuestros módulos y servicios desarrollados. Desde NetBeans podemos apreciar estos componentes desde la pestaña Services, con GlassFish en ejecución:

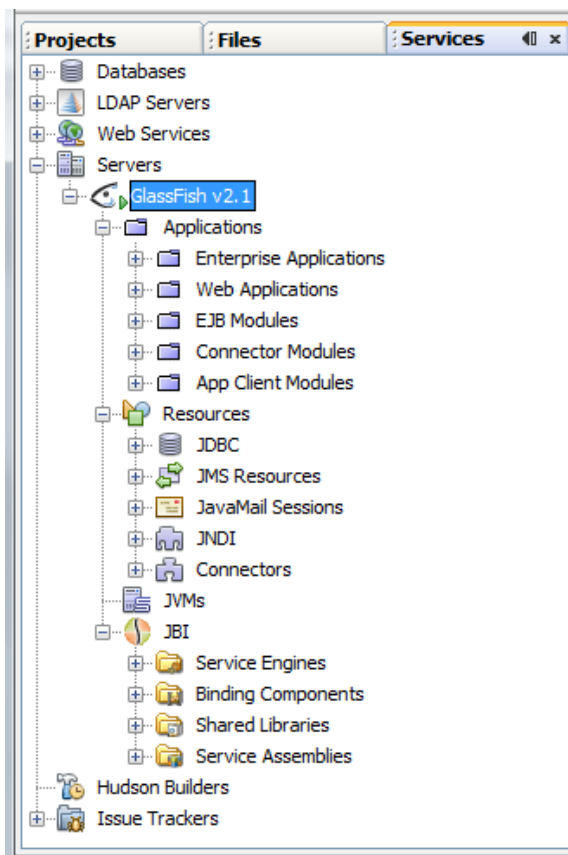


Figura C.11. OpenESB basado en Glassfish.

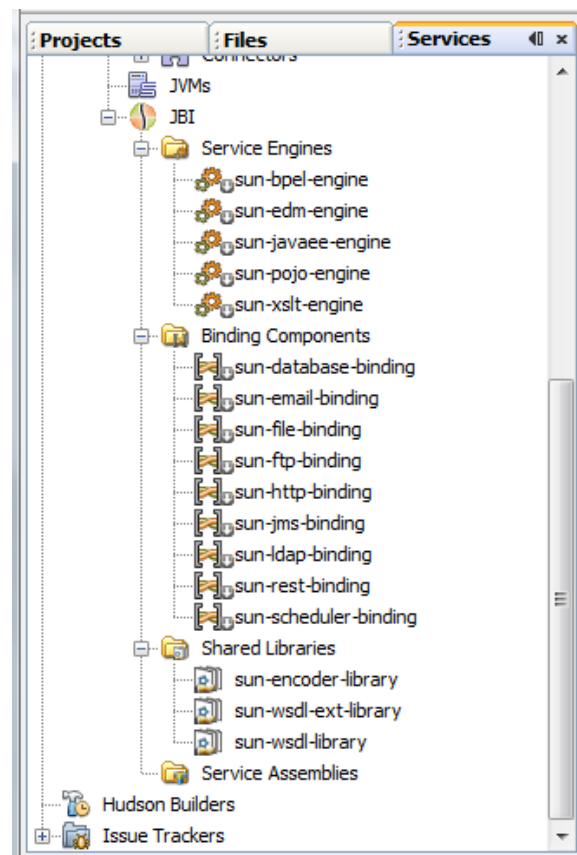


Figura C.12. OpenESB JBI.

A continuación la descripción de las siguientes categorías:

- **Service Engines.** Aquí se encuentran los motores de ejecución de los proyectos de la categoría SOA de NetBeans: BPEL Module (sun-bpel-engine), Data Mashup (sun-edm-engine), Intelligent Event Processing (sun-iep-engine), etc. Cada engine puede estar arrancado, detenido, desplegado o replegado.
- **Binding Components.** Son módulos orientados a la comunicación de componentes en JBI. Aportan transparencia al transporte de los mensajes en JBI respecto del protocolo de comunicación o infraestructura utilizado: HTTP (SOAP), FTP, colas, ficheros, bases de datos, email, ldap, etc. Existe todo un catálogo de componentes.



- **Shared Libraries.** Librerías comunes de las funcionalidades de los componentes.
- **Service Assemblies.** En esta categoría aparecerán nuestros proyectos, tras ser compilados, agregados a una Composite Application y desplegados.

De manera interactiva podemos invocar a JBI para que actúe sobre cada componente: su ciclo de vida y sus propiedades de configuración:

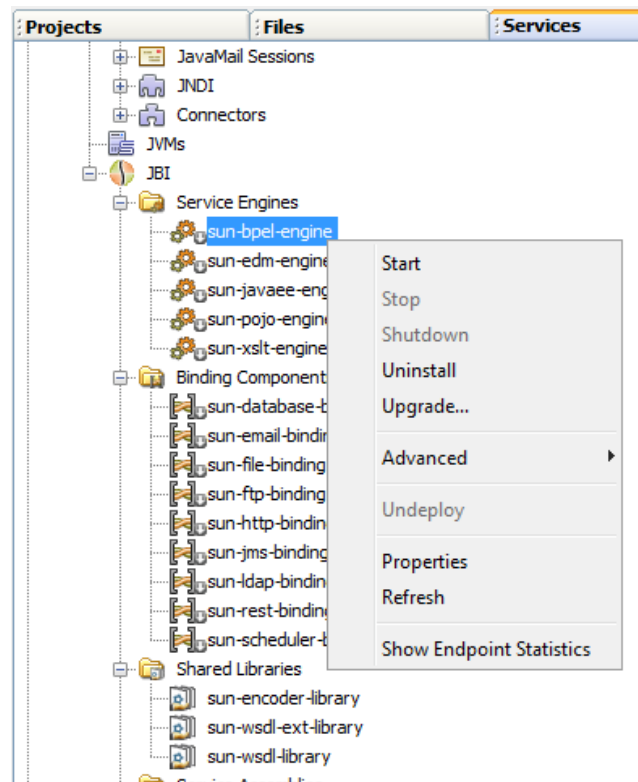


Figura C.13. Motor BPel en OpenESB.

Cada componente tiene asociados determinados recursos, como bases de datos, colas JMS, espacio en el filesystem, etc. Para conocer más a fondo esta configuración, podemos acceder a la consola de administración de Glassfish publicada en <http://localhost:4848/>. Se inicia sesión con el nombre de usuario y contraseña que establecimos en la instalación (admin / adminadmin)



Figura C.14. Interface de configuración OpenESB.

En el menú izquierdo se despliega el nodo JBI se pulsa sobre Components. Aparecerá una tabla con los instalados, y opciones para instalar o desinstalar otros:

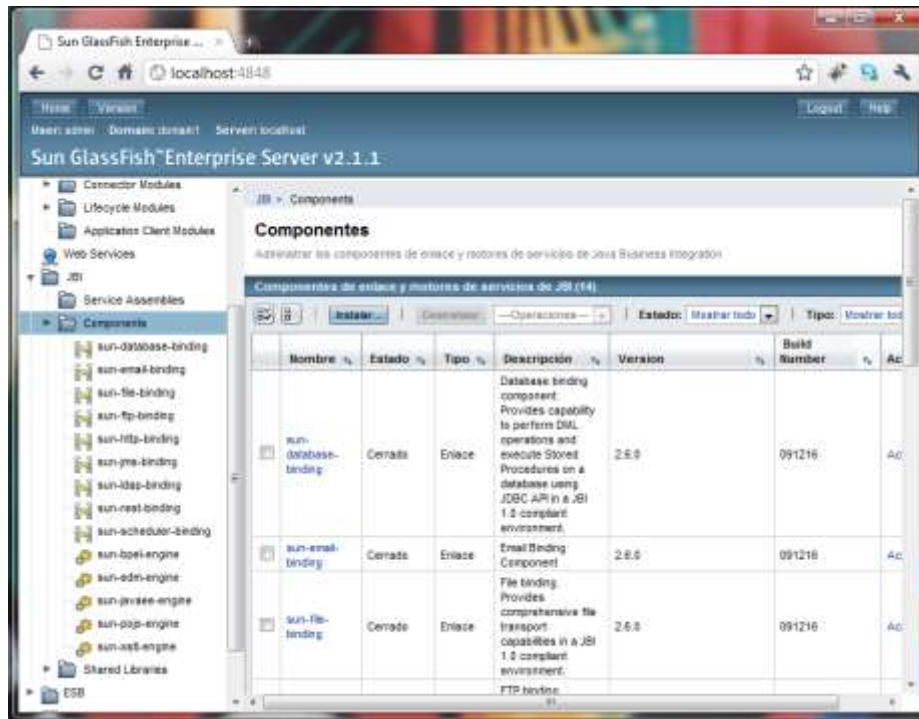


Figura C.15. Interface de Administración OpenESB Componentes JBI.

Pulsando sobre un componente en concreto accedemos a la edición de sus propiedades:

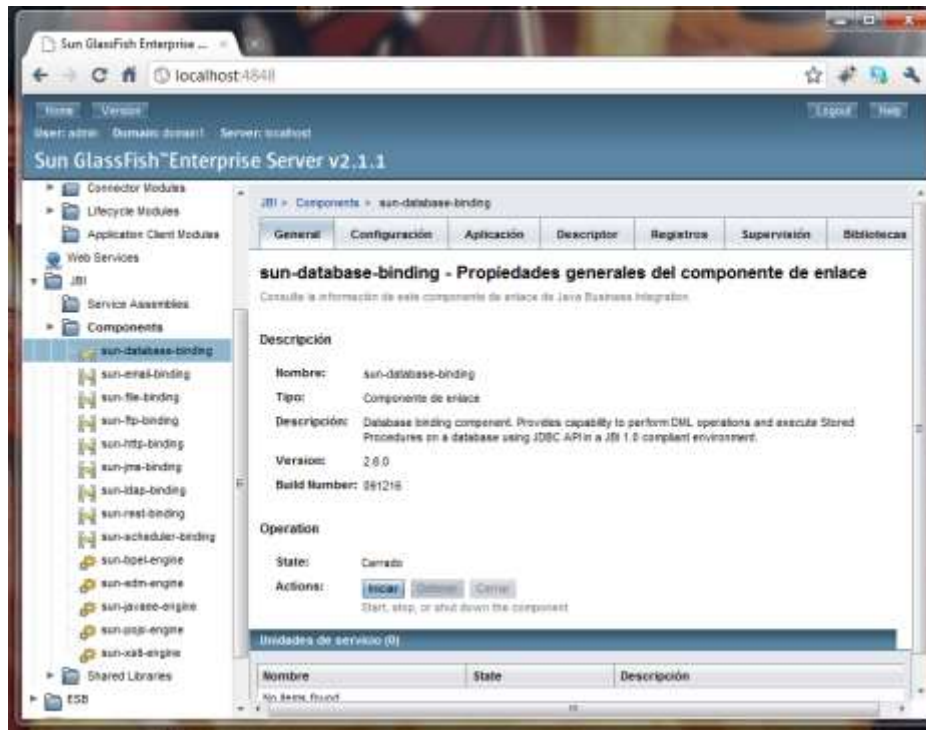


Figura C.16. Interface de administración OpenESB propiedades de un componente JBI

Para aquellos componentes con persistencia utilizan recursos JDBC basados en un pool de conexiones que pueden consultarse en el nodo Resources JDBC. Por ejemplo, el motor del gestor de eventos (IEP Service Engine, iepse) utilizará una base de datos Java DB (distribución Sun de Apache Derby) y tiene definidas dos pool de conexiones se puede examinar: DerbyPool:

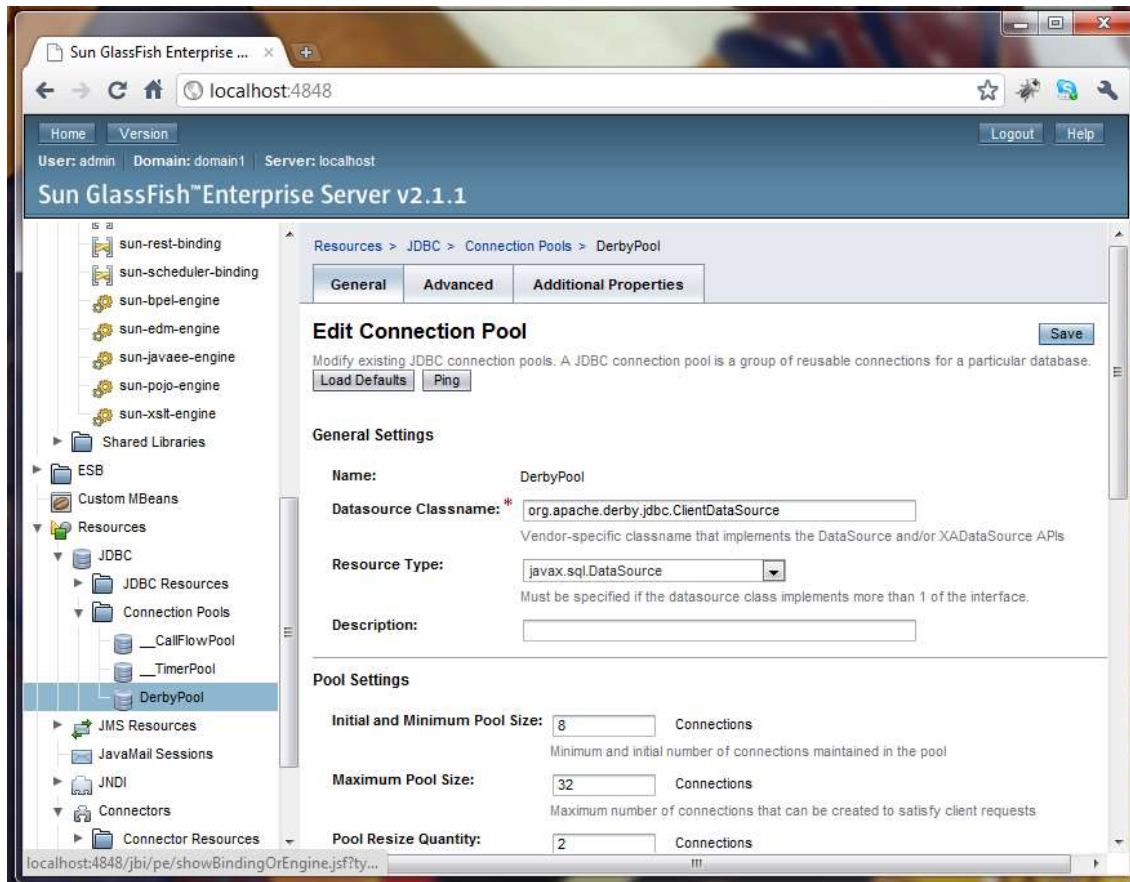


Figura C.17. Interface de Administración OpenESB Pool de Conexiones.

C.4. CONFIGURACIÓN DE POOL DE CONEXIONES POSTGRESQL.

Un pool de conexiones es un conjunto limitado de conexiones a una base, que es administrado por el servidor de aplicaciones del ESB (GlassFish), dichas conexiones pueden ser reutilizadas por los diferentes usuarios. Tanto las aplicaciones web como las aplicaciones de 3 capas utilizan este pool de conexiones, lo que las hace arquitecturas ventajosas frente a una arquitectura tradicional de cliente / servidor (2 capas).

El pool permite tener centralizado y controlado el manejo de las conexiones a la base de datos, ya que el acceso a la misma no se hace desde el cliente, como en una aplicación en 2 capas, sino que en este tipo de aplicación el acceso es realizado por el servidor de aplicaciones. Para ello podemos utilizar un pool de conexiones a la base de datos.

La cantidad de conexiones abiertas a una base de datos es limitada, dado que consumen muchos recursos del servidor de base de datos, y se requiere memoria y tiempo del procesador por cada nueva conexión. El manejo de un pool favorece la escalabilidad y performance de una aplicación.

En las aplicaciones cliente/servidor tradicionales, se mantiene una conexión a la base de datos por cada cliente final, esto es, si tenemos 100 clientes finales tendremos 100 conexiones a la base



de datos. Esto es aceptable para aplicaciones con un número de clientes relativamente pequeño, pero es inconveniente cuando la cantidad de clientes es grande, o impredecible (como en aplicaciones en Internet), por varias razones:

- La carga al servidor de base de datos es mayor cuantas más conexiones abiertas haya.
- En la mayoría de los DBMS se debe comprar una licencia de uso por cada usuario conectado.
- No es óptimo, dado que la mayoría de las conexiones están sin hacer nada una parte importante del tiempo.

La solución a estos problemas es permitir el compartir conexiones a la base de datos entre los diferentes clientes de la aplicación, de modo de que existan menos conexiones físicas a la base que clientes ejecutando la aplicación.

La idea general de un pool de conexiones, es que cada vez que un cliente necesita una conexión (esto es, cada vez que prepara un cursor, lo ejecuta, ejecuta una sentencia directamente, etc.), se la solicita al pool. Esta conexión pertenece al cliente hasta que de algún modo se determina que no la necesita más, y en ese momento se le puede asignar la conexión a otro cliente.

Si un cliente necesita una conexión y no existe ninguna disponible, se queda esperando hasta que alguna conexión se libere. Otra ventaja es que las conexiones se mantienen abiertas, por lo que los clientes se ahorran el tiempo de conexión al DBMS al iniciar la aplicación.

Creamos la base de datos soa_sis y para el rol del mismo nombre soa_sis.

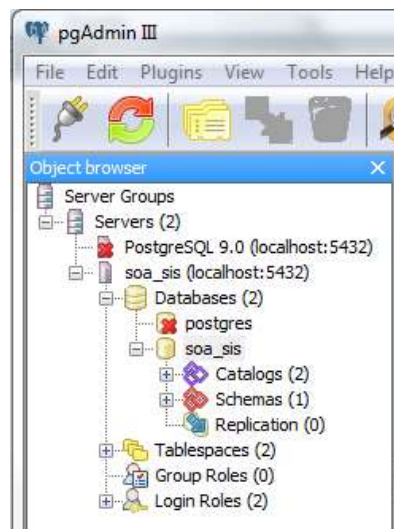


Figura C.18. Base datos postgresQL

En la versión de Netbeans 6.7.1 ya dispone de conexión a Oracle por Jdbc como se muestra en la Figura C.19. En caso de no disponer de él se debe agregar y agregar donde click derecho en Databases, Drivers, New Driver y se le da la ruta donde quedo el archivo driver Jdbc.

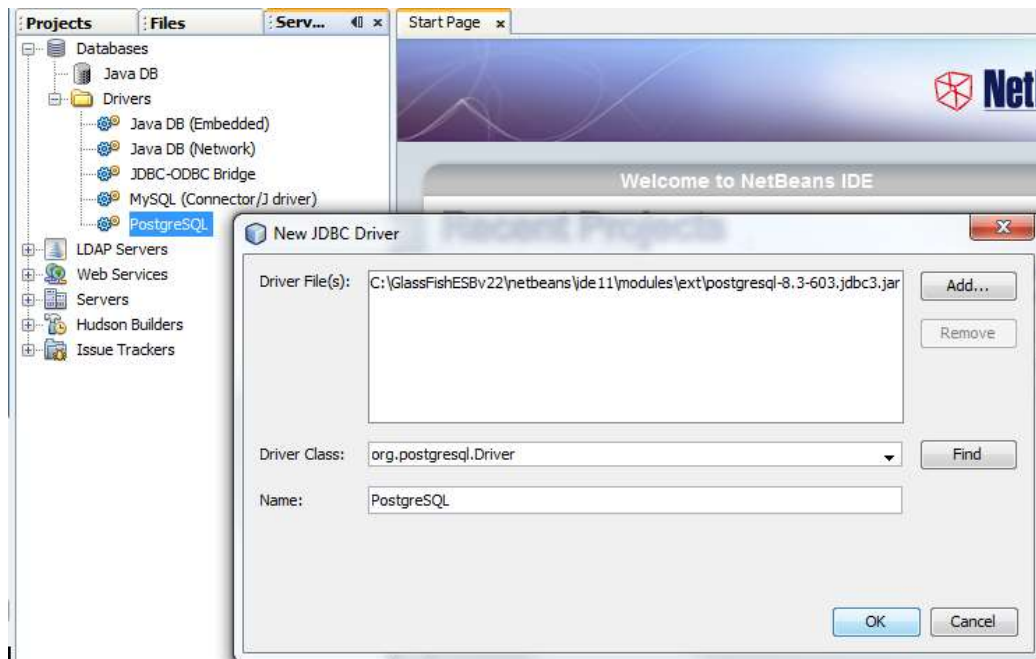


Figura C.19. Drives JDBC PostgreSQL.

Ahora lo que se crea una conexión que servirá de referencia para poder crear el pool de conexiones, en la lista mostrada anteriormente, se da click derecho a PostgreSQL y se selecciona la opción “Connect Using” y aparecerá una ventana en la se llena todos los datos requeridos, quedando algo como esto:

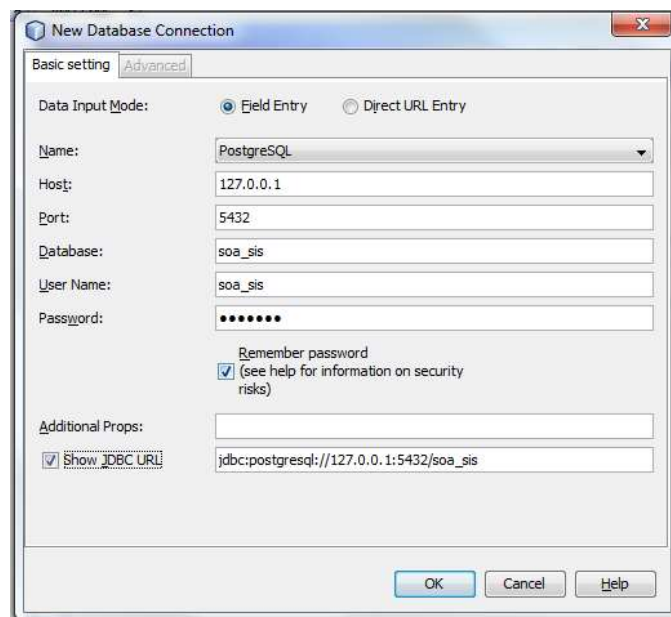


Figura C.20. Agrgar driver Jdbc PostgreSQL

Se da click en OK y luego aparecerá una ventana en donde se debe seleccionar el esquema, entonces debido a que se está utilizando el esquema por defecto entonces se debe desplegar el



combo y seleccionar “public” y luego se verá la referencia de conexión dentro de la lista como se muestra en la imagen:

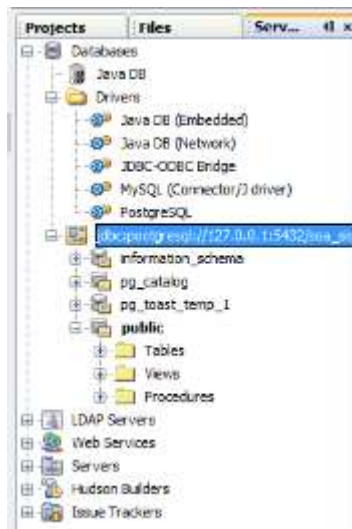


Figura C.21. Conexión JDBC PostgreSQL.

A continuación en un proyecto web se le da click derecho y *New*, a continuación la opción *Other* y luego dentro de las opciones ubicarse en *Glassfish*, quedando de esta manera:

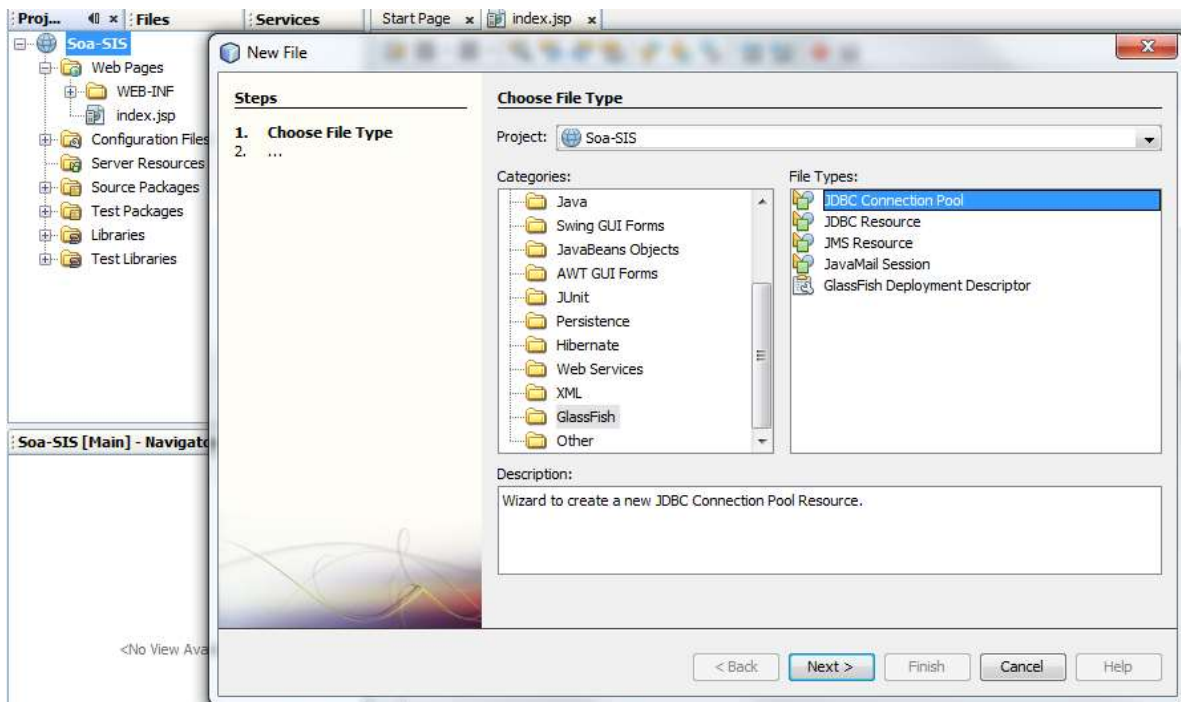


Figura C.22. Crear un Pool de conexiones.

Una vez aquí se hacen dos procesos, el primero será seleccionar la opción *JDBC Connection Pool*, luego click en *Next* y en la siguiente ventana colocar como nombre *PoolSOA-SIS* y en la



relación de conexiones existentes, seleccionar la conexión activa que se había registrado anteriormente tal como se muestra en la figura:

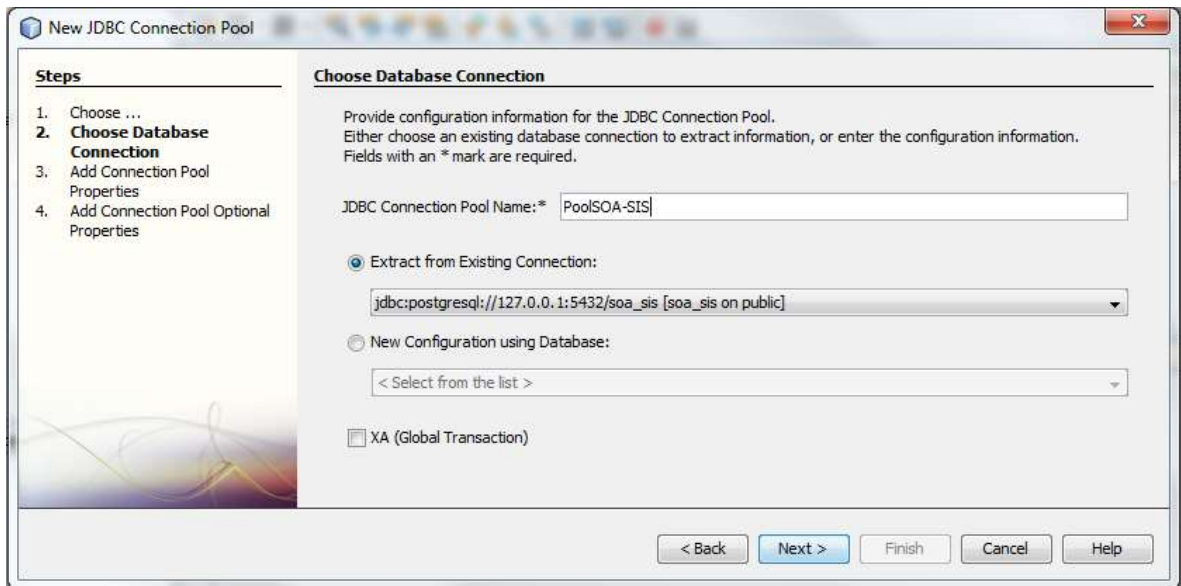


Figura C.23. Seleccionar conexión para crear el Pool de conexiones.

Luego en la siguiente ventana se muestran los datos de la conexión pero se debe agregar: `serverName`, `databaseName` y `portNumber`. Como se muestra a continuación:

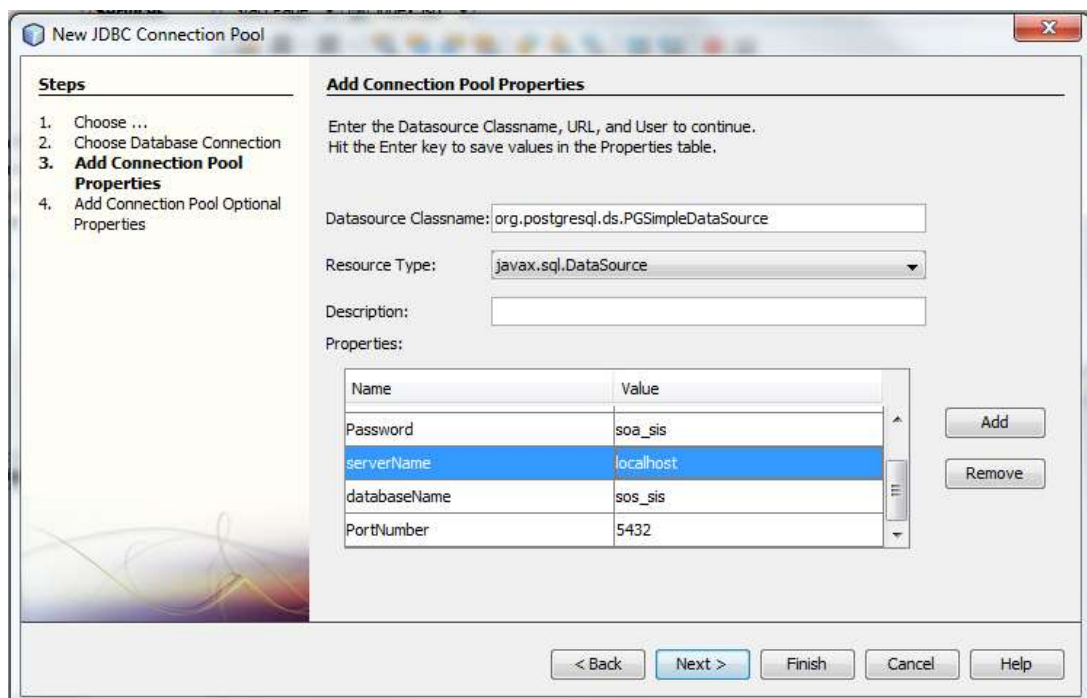


Figura C.24. Configuración del Pool de conexiones.

Cuando esto termina se puede ver en el proyecto el archivo `sun-resources.xml`, como se muestra:

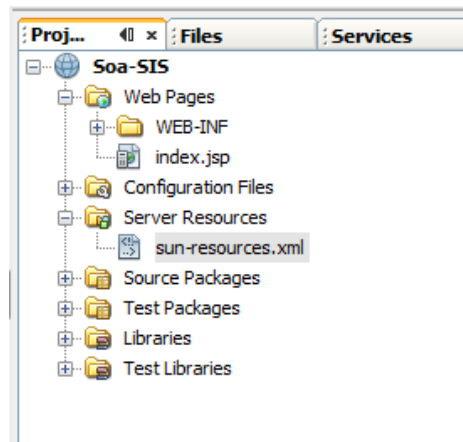


Figura C.25. Archivo de configuración de Pool.

Este será su contenido:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE resources PUBLIC "-//Sun Microsystems, Inc.//DTD Application Server 9.0 Resource Definitions //EN"
"http://www.sun.com/software/appserver/dtds/sun-resources_1_3.dtd">
<resources>
  <jdbc-connection-pool allow-non-component-callers="false" associate-with-thread="false" connection-creation-retry-
attempts="0" connection-creation-retry-interval-in-seconds="10" connection-leak-reclaim="false" connection-leak-
timeout-in-seconds="0"
                    connection-validation-method="auto-commit"
                    datasource-
classname="org.postgresql.ds.PGSimpleDataSource" fail-all-connections="false" idle-timeout-in-seconds="300" is-
connection-validation-required="false" is-isolation-level-guaranteed="true" lazy-connection-association="false" lazy-
connection-enlistment="false" match-connections="false" max-connection-usage-count="0" max-pool-size="32" max-
wait-time-in-millis="60000" name="PoolSOA-SIS" non-transactional-connections="false" pool-resize-quantity="2" res-
type="javax.sql.DataSource" statement-timeout-in-seconds="-1" steady-pool-size="8" validate-atmost-once-period-in-
seconds="0" wrap-jdbc-objects="false">
    <property name="URL" value="jdbc:postgresql://127.0.0.1:5432/soa_sis"/>
    <property name="User" value="soa_sis"/>
    <property name="Password" value="soa_sis"/>
    <property name="serverName" value="localhost"/>
    <property name="databaseName" value="sos_sis"/>
    <property name="PortNumber" value="5432"/>
  </jdbc-connection-pool>
</resources>
```

Luego queda agregar el recurso o referencia que utilizara el pool para ser llamado desde una clase que necesite la conexión, para esto se le click derecho al proyecto, seleccionar *New* luego *Other* y dentro de la ventana de opciones seleccionar "Glassfish" y luego seleccionar la opción *JDBC Resource* dar click en *Next* y aparecerá una ventana en donde se selecciona el pool creado anteriormente y en la opción *JNDI Name* colocar como nombre "jdbc/soa_pool" quedando de la siguiente manera:

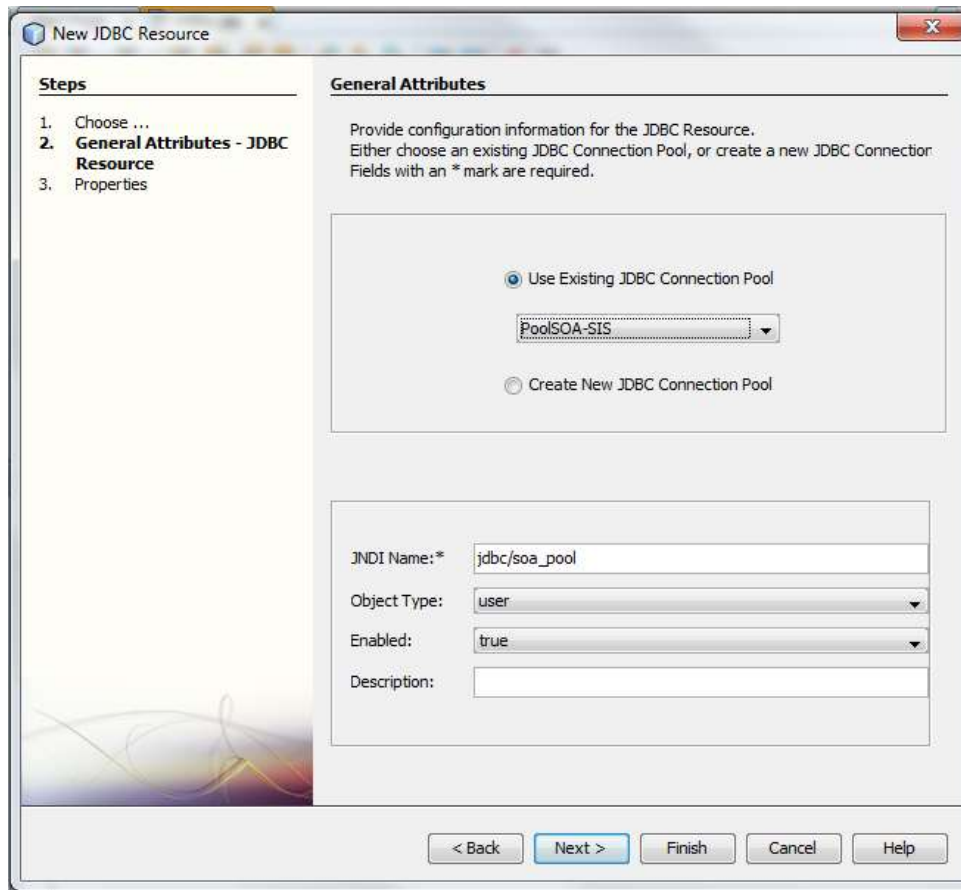


Figura C.26. Configurar recurso en de Pool en Glasfish.

De esta forma el archivo queda:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE resources PUBLIC "-//Sun Microsystems, Inc.//DTD Application Server 9.0 Resource Definitions //EN"
"http://www.sun.com/software/appserver/dtds/sun-resources_1_3.dtd">
<resources>
  <jdbc-resource enabled="true" jndi-name="jdbc/soa_pool" object-type="user" pool-name="PoolSOA-SIS">
    <description/>
  </jdbc-resource>
  <jdbc-connection-pool allow-non-component-callers="false" associate-with-thread="false" connection-creation-retry-
attempts="0" connection-creation-retry-interval-in-seconds="10" connection-leak-reclaim="false" connection-leak-
timeout-in-seconds="0" connection-validation-method="auto-commit" datasource-
classname="org.postgresql.ds.PGSimpleDataSource" fail-all-connections="false" idle-timeout-in-seconds="300" is-
connection-validation-required="false" is-isolation-level-guaranteed="true" lazy-connection-association="false" lazy-
connection-enlistment="false" match-connections="false" max-connection-usage-count="0" max-pool-size="32" max-
wait-time-in-millis="60000" name="PoolSOA-SIS" non-transactional-connections="false" pool-resize-quantity="2" res-
type="javax.sql.DataSource" statement-timeout-in-seconds="-1" steady-pool-size="8" validate-atmost-once-period-in-
seconds="0" wrap-jdbc-objects="false">
  <property name="URL" value="jdbc:postgresql://127.0.0.1:5432/soa_sis"/>
  <property name="User" value="soa_sis"/>
  <property name="Password" value="soa_sis"/>
  <property name="serverName" value="localhost"/>
</jdbc-connection-pool>
</resources>
</pre>
```



```
<property name="databaseName" value="sos_sis"/>
<property name="PortNumber" value="5432"/>
</jdbc-connection-pool>
</resources>
```

Se da deploy a la aplicación. Con esto se genera el pool de conexiones y nombre de recurso asignado a ese pool en nuestro servidor de aplicaciones. Esto se puede verificar entrando a la consola de administración de Glassfish; una vez dentro de ella ubicarse en la opción Recursos, luego en JDBC y luego para ver el pool creado ubicarse en Conjunto de Conexiones, y luego dentro de la lista debería estar *Pool/SOA-SIS*, lo seleccionamos y nos ubicamos en la parte derecha del detalle y hacemos clic en “Sondeo”, si todo marcha bien deberías tener lo siguiente:

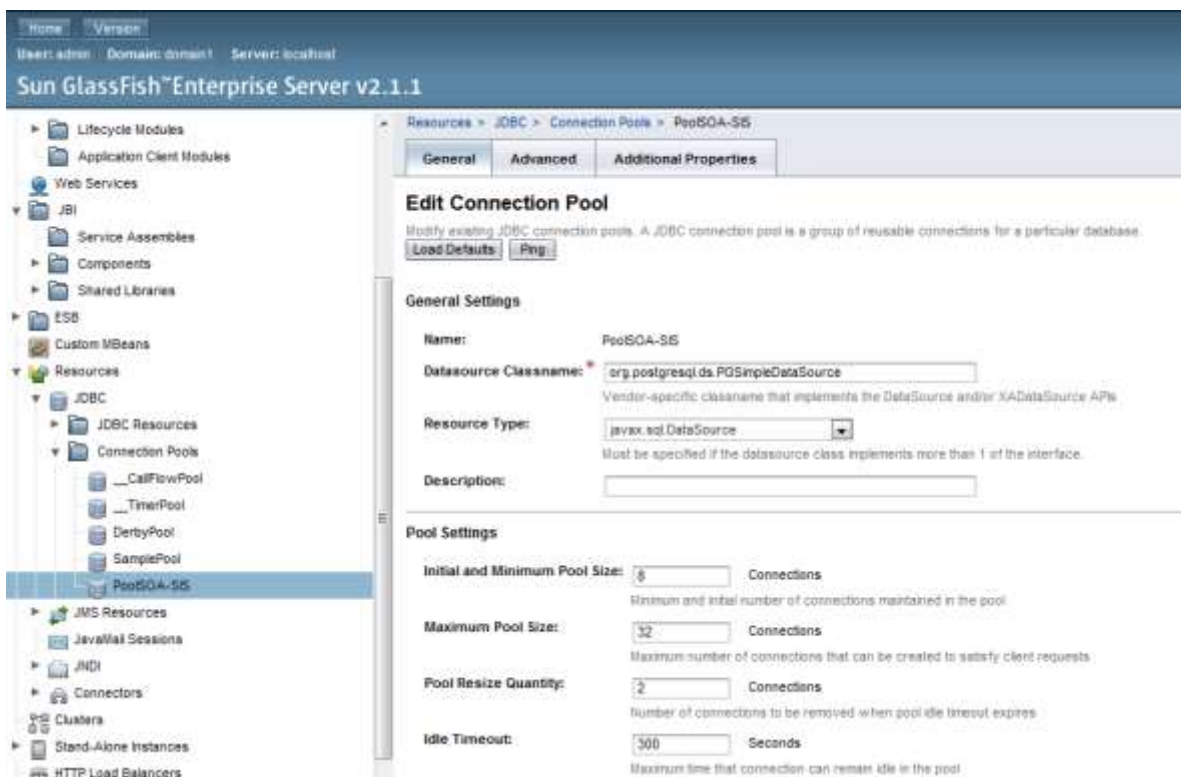


Figura C.27. Pool de conexiones PostgreSQL.

Como se puede ver en la figura por defecto el pool establece 32 conexiones dedicadas y en condiciones de arranque solicita 8.



D INSTALACIÓN DE HL7 BUILDING COMPONENT (BC)

Se describirá el proceso para la configuración de HL7 Building component en GlassFish 2.2 para integrarlo en los proyectos que requieran transformación de mensajes.

D.1. PRERREQUISITOS

Se descarga de la siguiente url <http://download.java.net/jbi/binaries/open-esb-full-install/nbm/latest/>, el codificador HL7, el módulo de diseño para Netbeans y la extensión para la publicación de mensajes en HL7.

- com-sun-encoder-hl7.nbm
- org-netbeans-modules-encoder-hl7-aip.nbm
- org-netbeans-modules-wsdlexensions-hl7.nbm

Descargar el codificador encoderlib.jar disponible en:

<http://download.java.net/jbi/binaries/open-jbi-components/main/nightly/latest/ojc/>

También se debe obtener los esquemas del formato HL7 de:

<http://wiki.open-esb.java.net/attach/HL7/hl7.zip>

D.1.1. INSTALACIÓN DE HL7 ENCODER

Se debe reemplazar la versión del codificador que por defecto trae el ESB, dando click en la librería sun-encoder-library y uninstall.

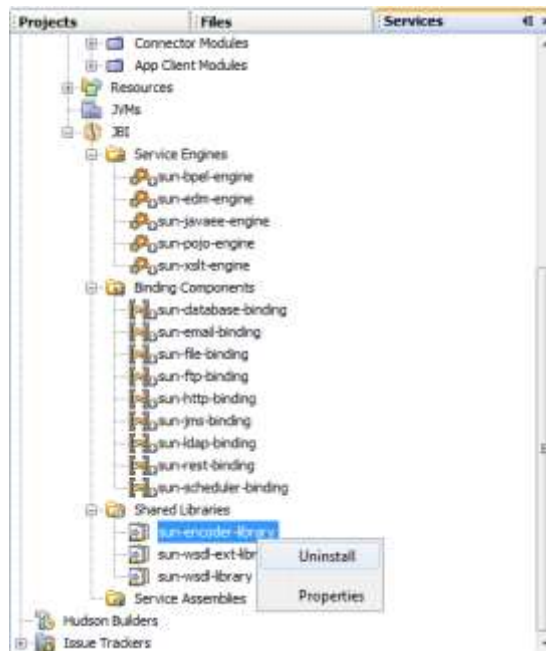


Figura D.1. Desinstalación del codificador.

Dando click derecho sobre las librerías compartidas accedemos a la opción de instalar

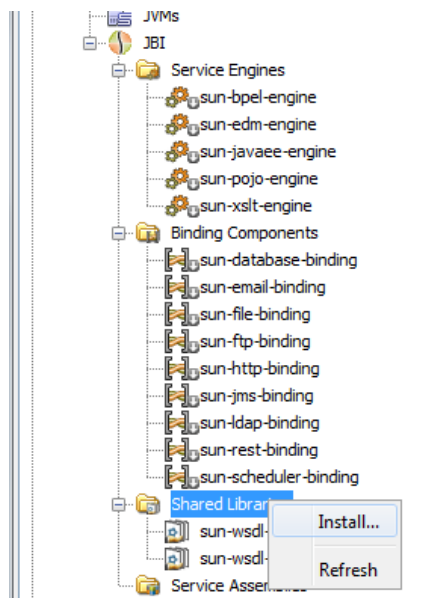


Figura D.2. Instalación del codificador HL7.

Se debe agregar la librería descargada

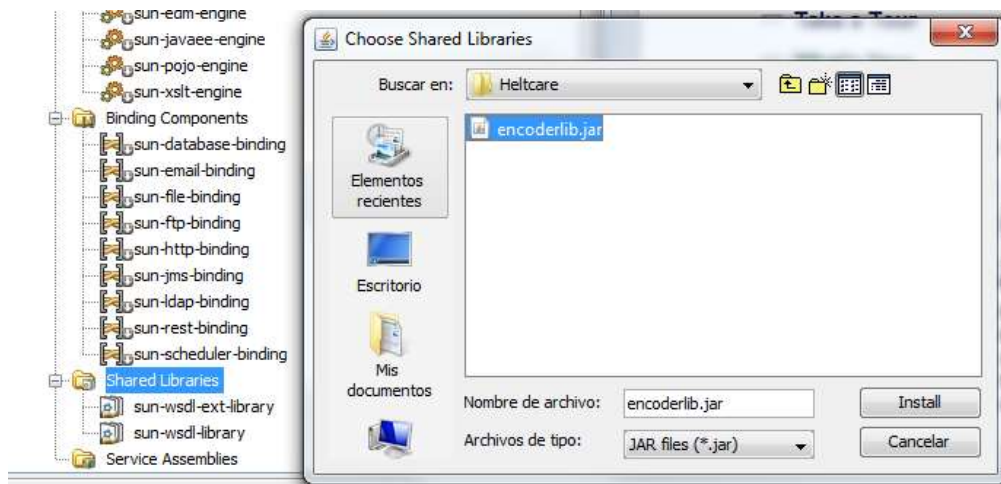


Figura D.3. Selección de las librerías que componen el codificador.

De esta forma al dar click derecho sobre la librería de codificación y ver las propiedades debe resultar algo como:

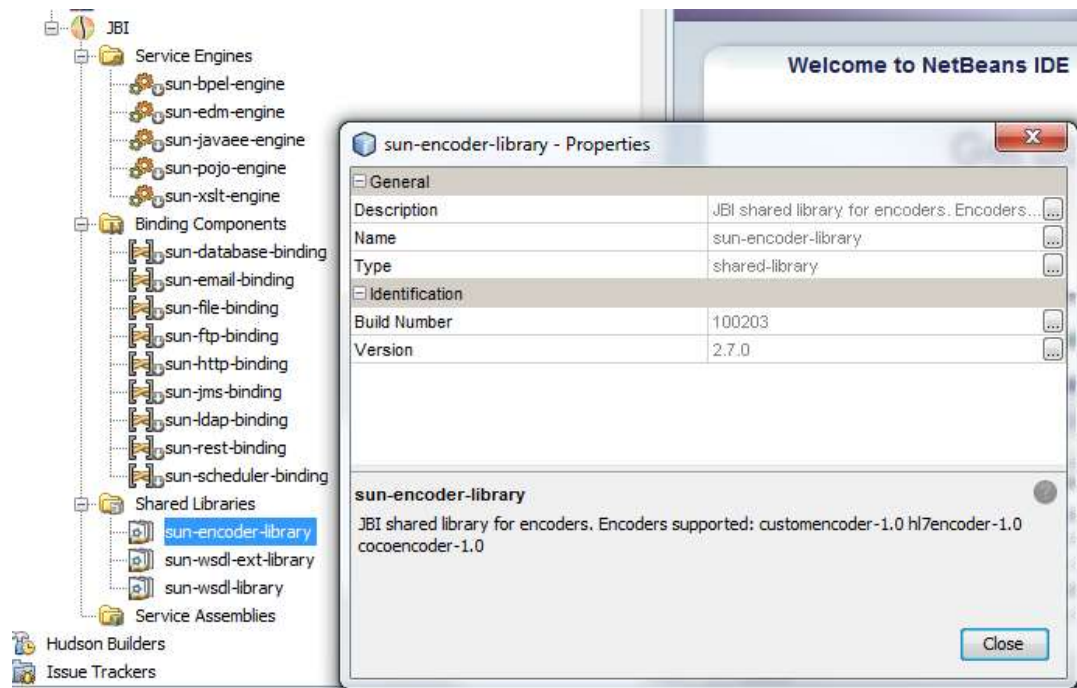


Figura D.4. Propiedades del codificador HL7.

D.1.2. INSTALACIÓN DE PLUGINS PARA NETBEANS IDE

Se procede a instalar los plugins de netbeans, dando en el menú Tools > Plugins y en la pestaña downloaded se da click en el botón Add Plugins.

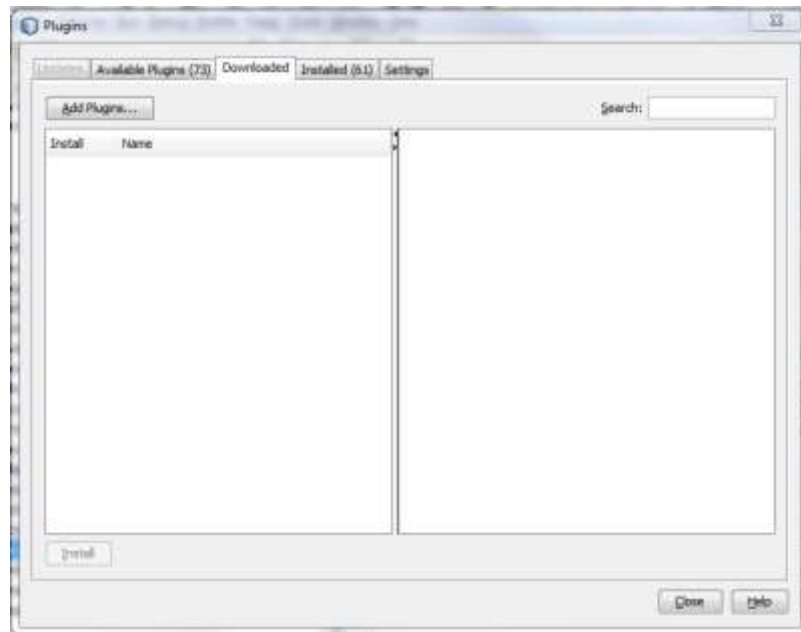


Figura D.5. Netbeans IDE - Administración de complementos.

Y se agregan los plugins descargados inicialmente.

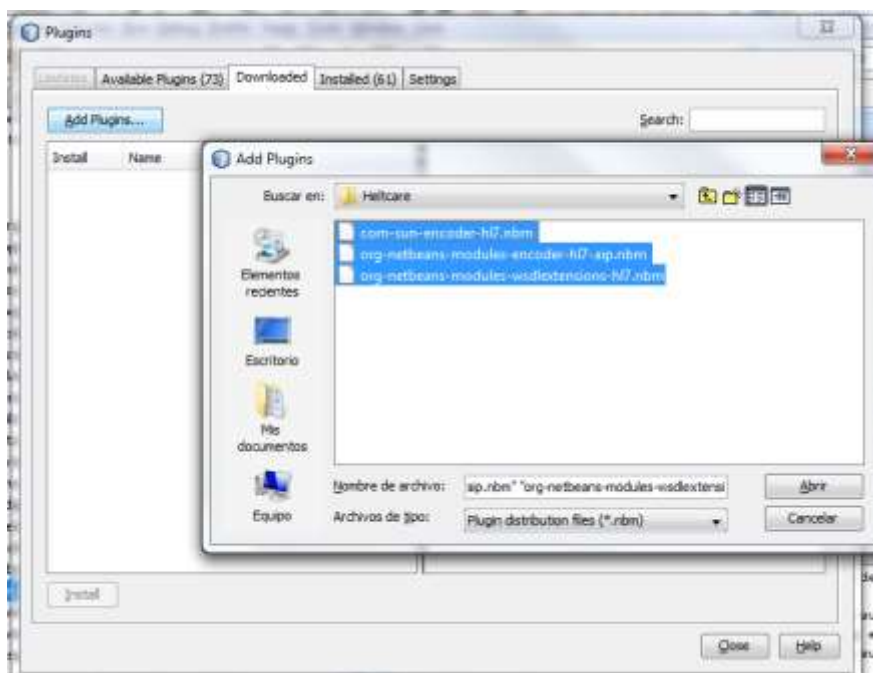


Figura D.6. Netbeans IDE - Agregar plugins

Se verifica la correcta agregación de los componentes y se pulsa el botón Install (parte inferior), de esta forma se inicia el instalador de plugins.

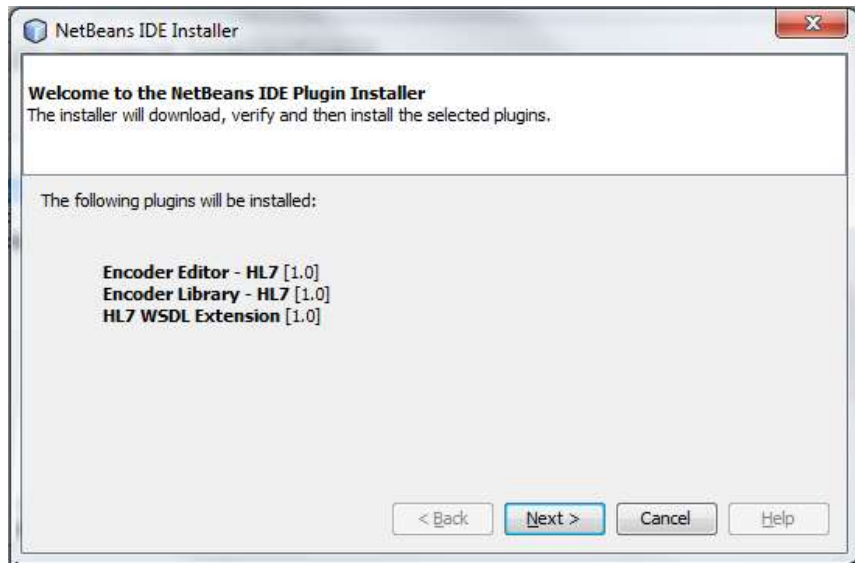


Figura D.7. Netbeans IDE - Asistente de configuración de Plugins.

Se acepta los términos de la licencia, de esta forma nos perdira que confirmemos la instalación dado que no están firmados digitalmente.

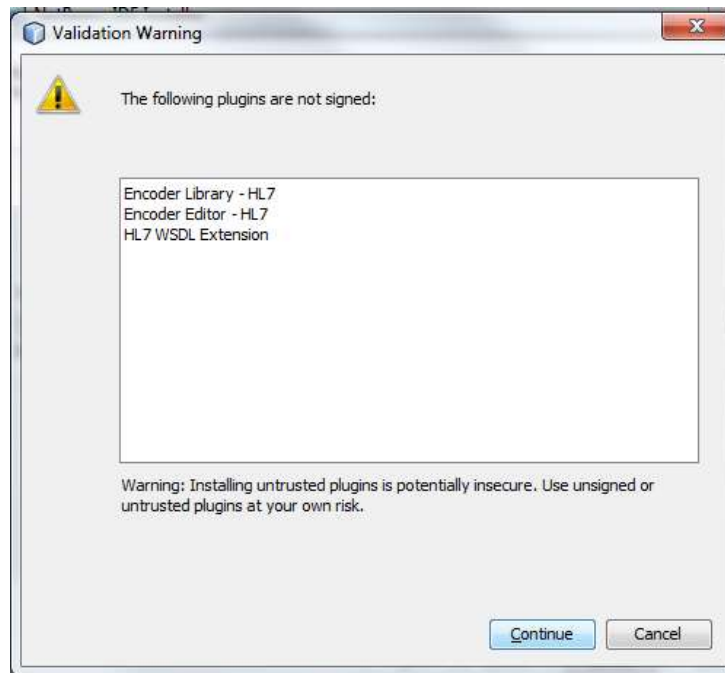


Figura D.8. Netbeans IDE - Mensaje de validación de firmas digitales.

Se pulsa el botón Continuar, de esta forma la instalación continua y damos reiniciar el IDE.

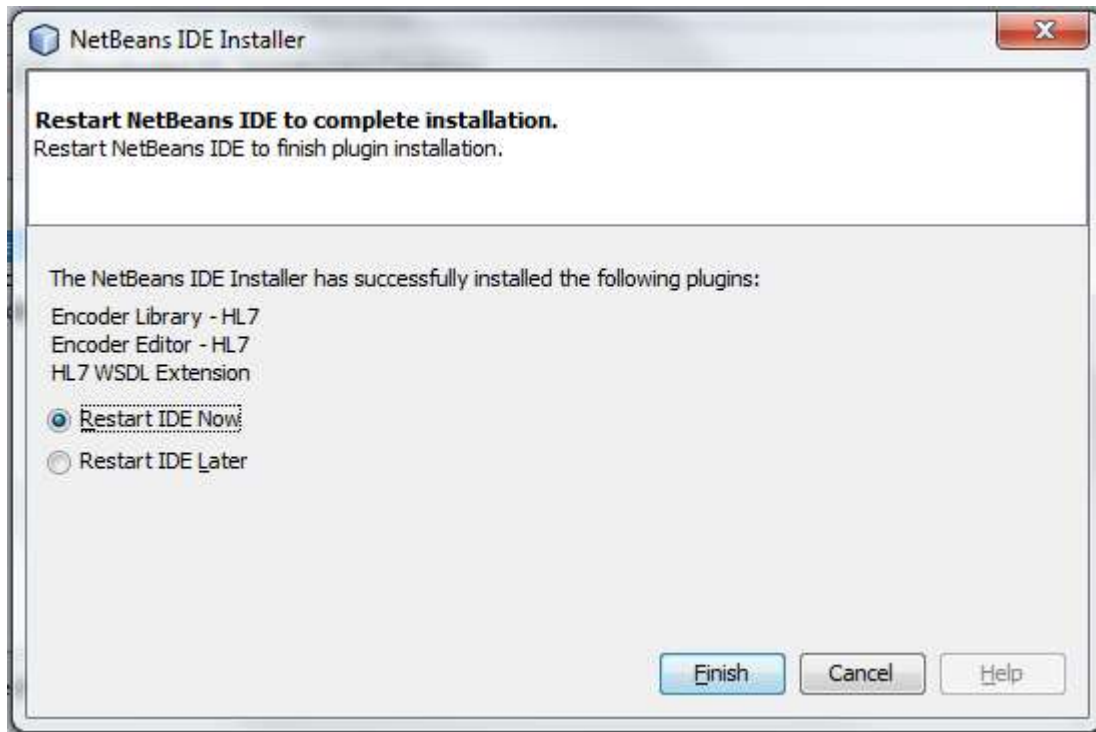


Figura D.9. Netbeans IDE - Instalación completa de Plugins.

Una vez reiniciado el IDE podemos verificar la correcta inclusión de los elementos en el servidor de aplicaciones y en Netbeans dando click en el menú Tools > Plugins y en la pestaña installed.

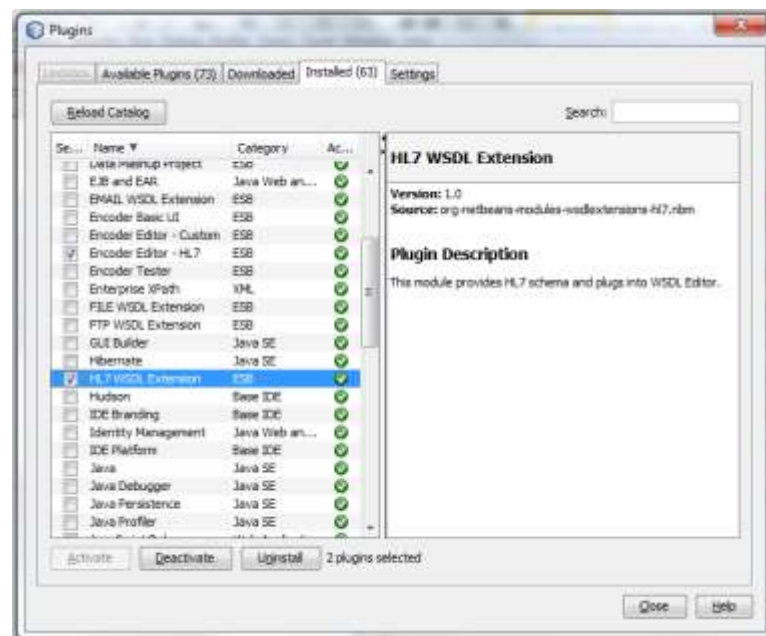


Figura D.10. Netbeans IDE - Plugins instalados.

De esta forma se puede decir que las dependencias están listas para la instalación de HL7 Building component.



D.2. INSTALACIÓN DE HL7 BUILDING COMPONENTS

Se debe descargar el componente de desarrollo HL7 BC (hl7bc.jar) del siguiente URL:

<http://download.java.net/jbi/binaries/open-jbi-components/branches/soarelease/nightly/latest/ojc/>

Luego de iniciar el servidor GlassFish al desplegar en el módulo de Java Business Integration (JBI) en los componentes en se puede dar click derecho en bulding componets se puede acceder a instar e iniciar.

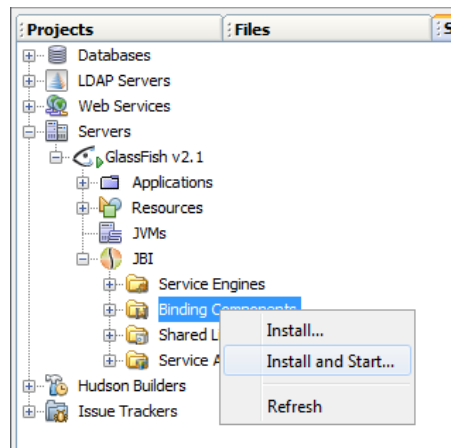


Figura D.11. Instalación de HL7 BC

Se debe agregar el componente desde el directorio de la descarga

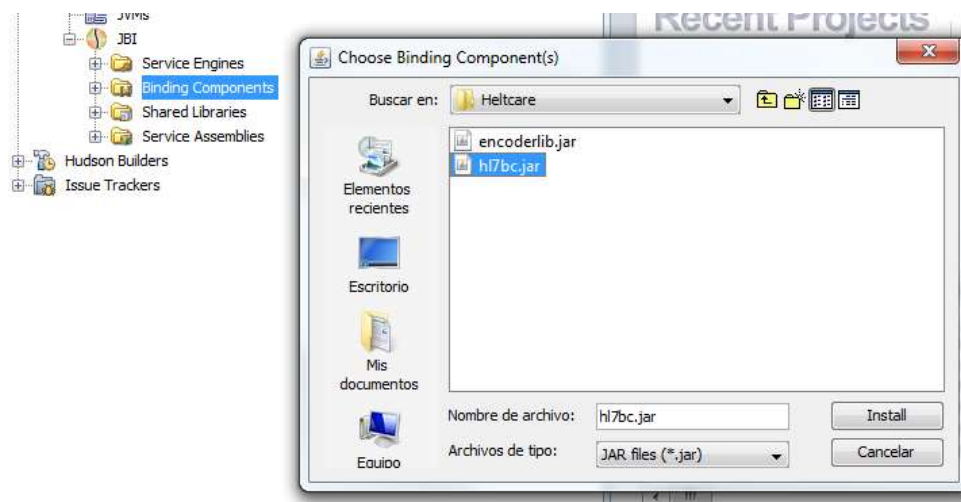


Figura D.12. Agregar componente HL7

Cuando despliega el asistente de configuración se da click en Install.

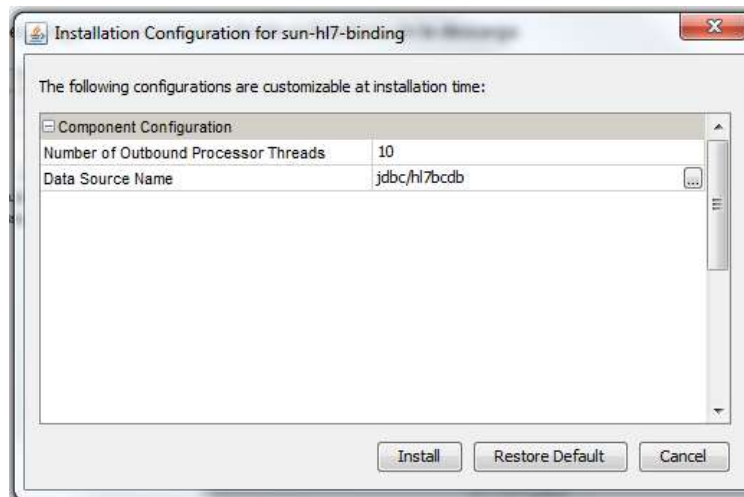


Figura D.13. Propiedades del componente HL7.

De esta forma el componente ha sido instalado.

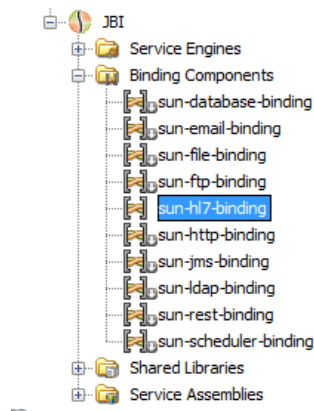


Figura D.14. Componente HL7 agregado.

D.3. CREACIÓN Y VALIDACIÓN DE UN WSDL EMPLEANDO HL7 ENCODER PARA CREACIÓN DE UN MENSAJE HL7.

Dando click en nuevo Proyecto y la opción SOA seleccionar BEL Module.

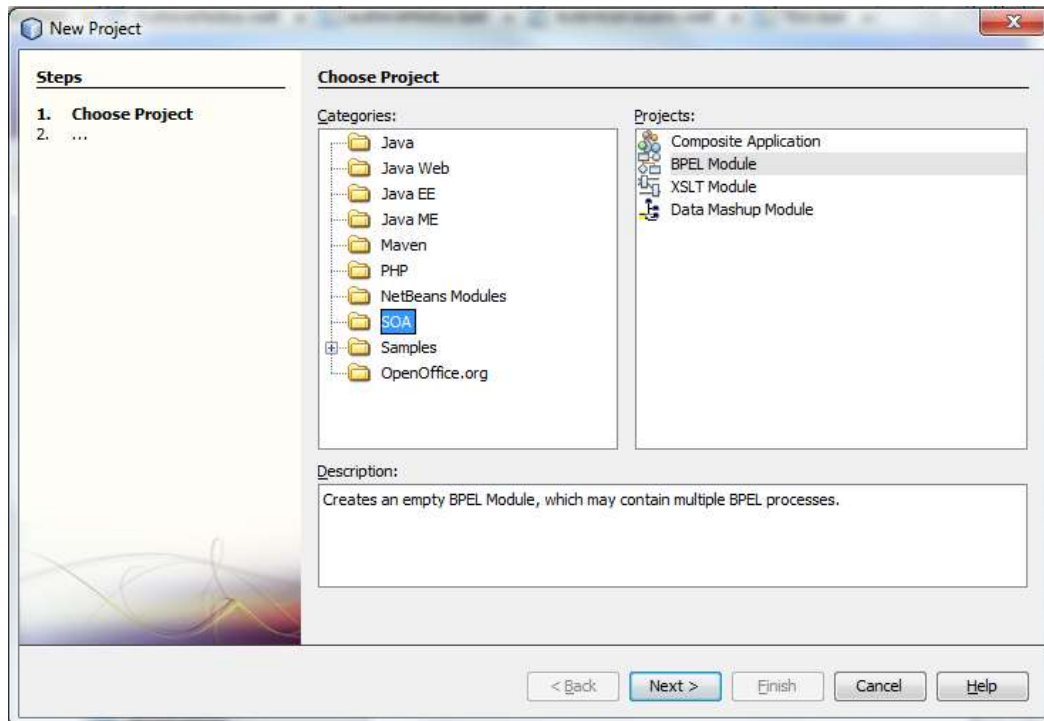


Figura D.15. Nuevo Proyecto HL7 Building component.

Le asignamos un nombre al modulo.

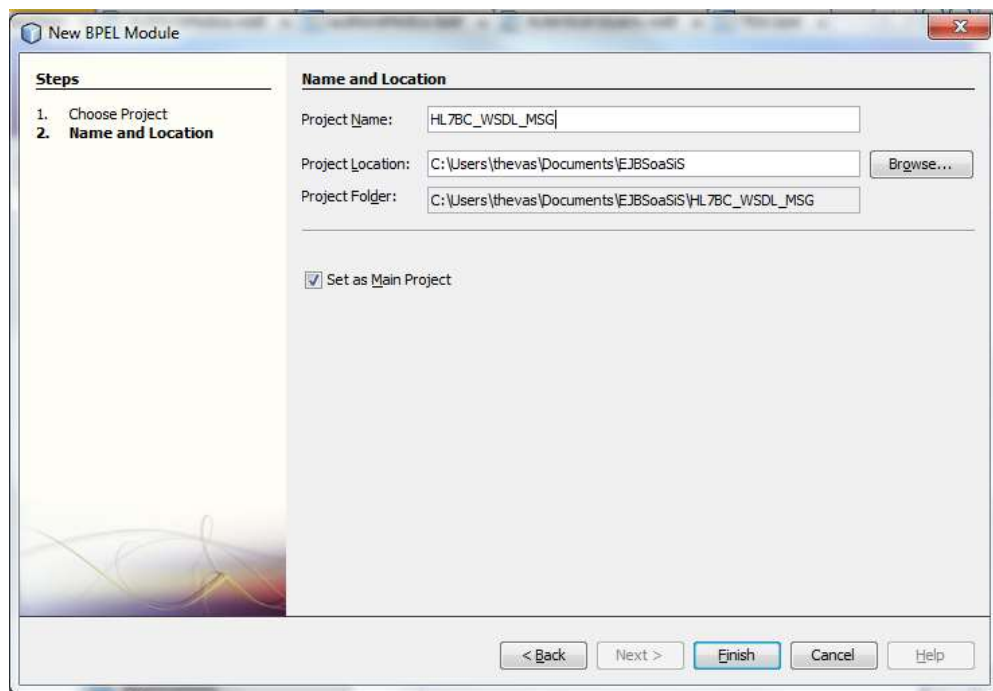


Figura D.16. Nombre del Modulo HL7 BC.

Ahora se crean un nuevo archivo de esquema XML, estos documentos descartados contienen la estructura del documento HL7 que se creará.

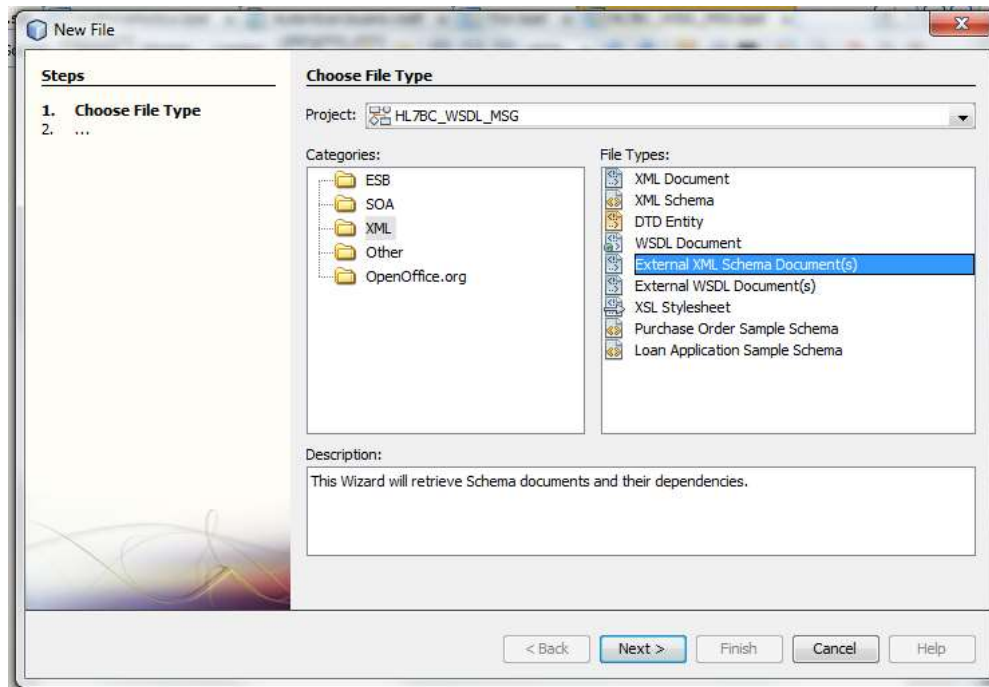


Figura D.17. Nuevo Esquema XML.

Posteriormente seleccionamos la ruta del documento que queremos obtener.

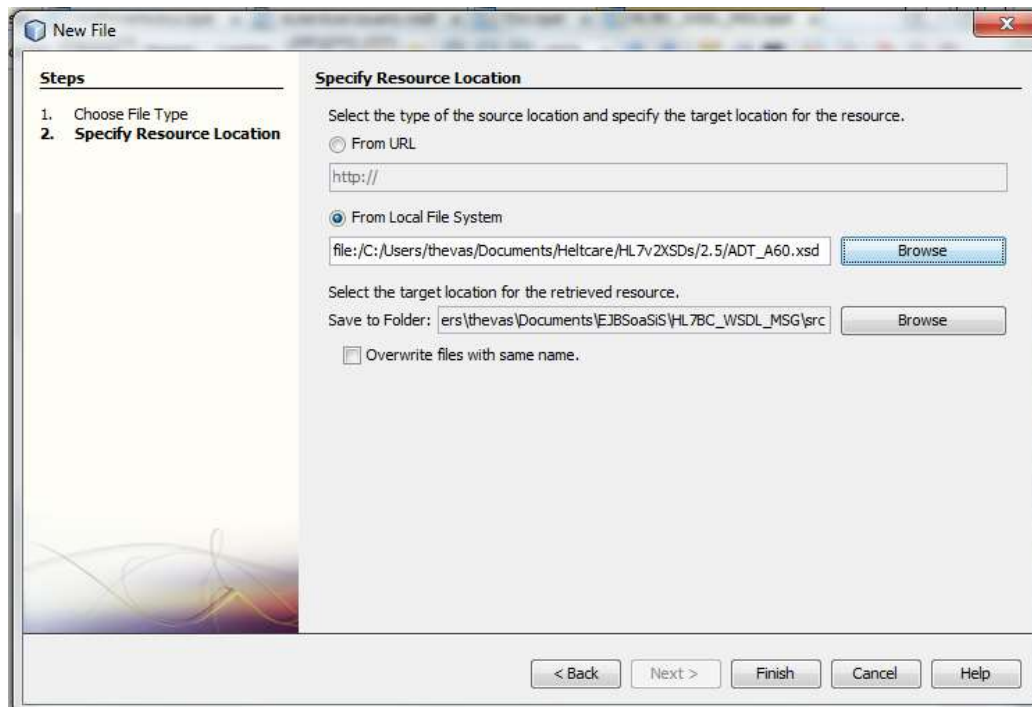


Figura D.18. seleccion de la estructura del mensaje HL7.

Ahora se debe definir la WSDL que representa el servicio de parseo dando click en nuevo WSDL document.

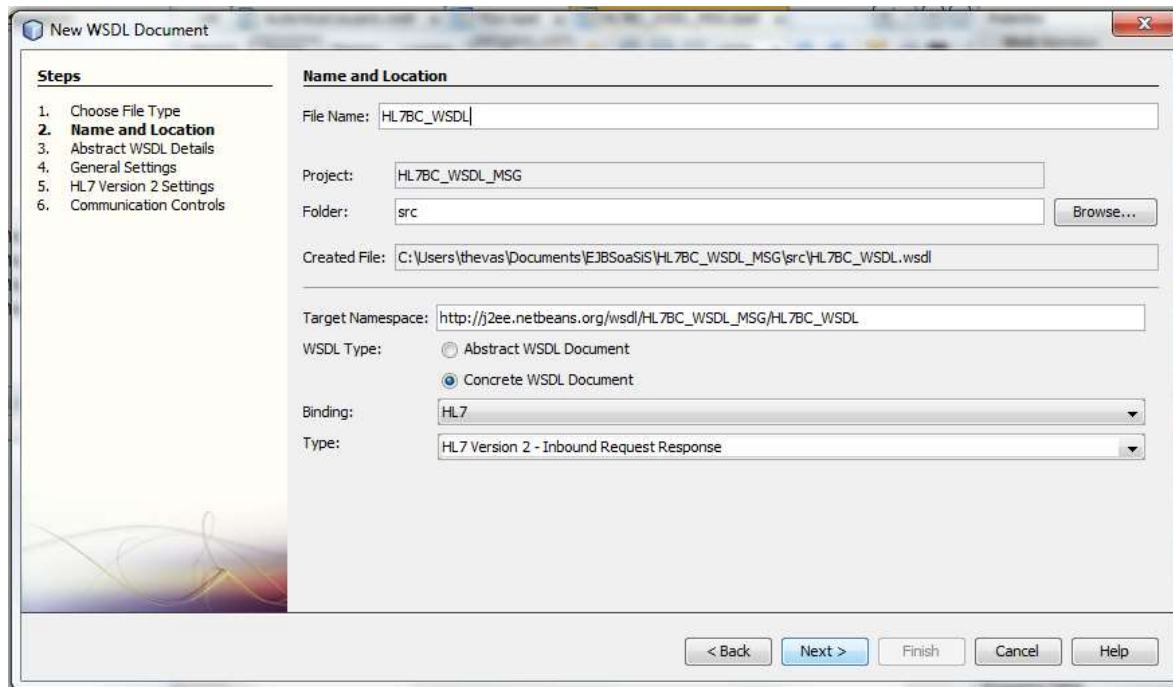


Figura D.19. Nombre de HL7BC_SWDL.

Le datos que sea un aservicio HL7 de solicitud y respuesta como se muestra en la figura D.19.

A continuación se le da el tipo de formato de respuesta que seda para ello se le asigna XSD XHL7 ().

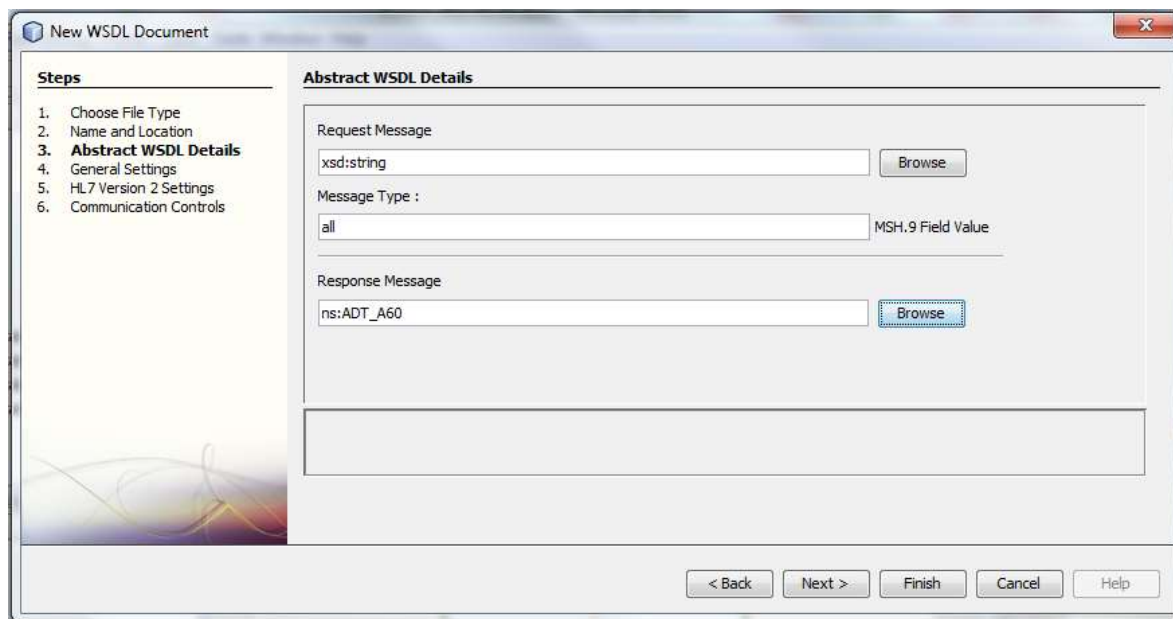


Figura D.20. Tipo de respuesta del servicio.

A continuación seleccionamos el codificador de HL7 que se empleara como mestra la figura D.21.

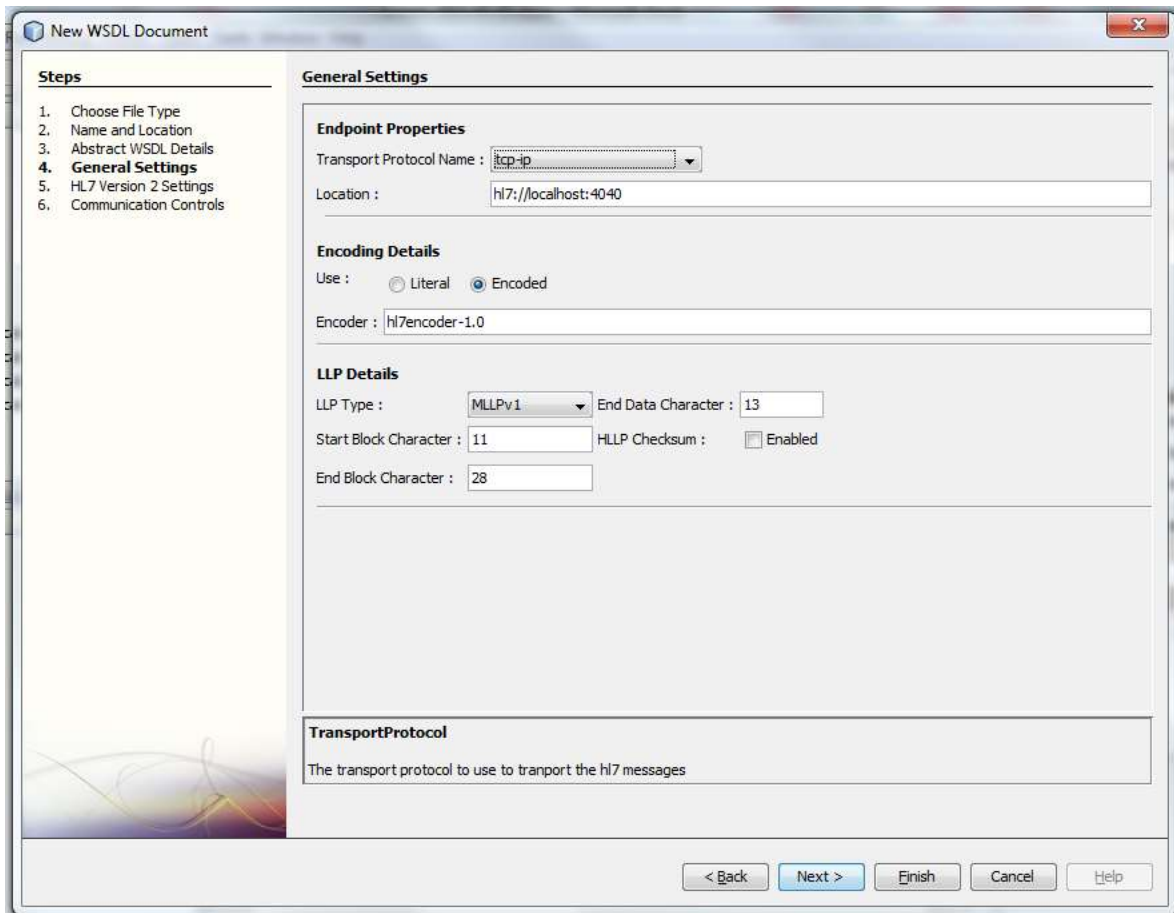


Figura D.21. Selección de codificador y el protocolo de mensajería.

Luego se configuran propiedades particulares del protocolo HL7 v2 dejamos por default los valores.

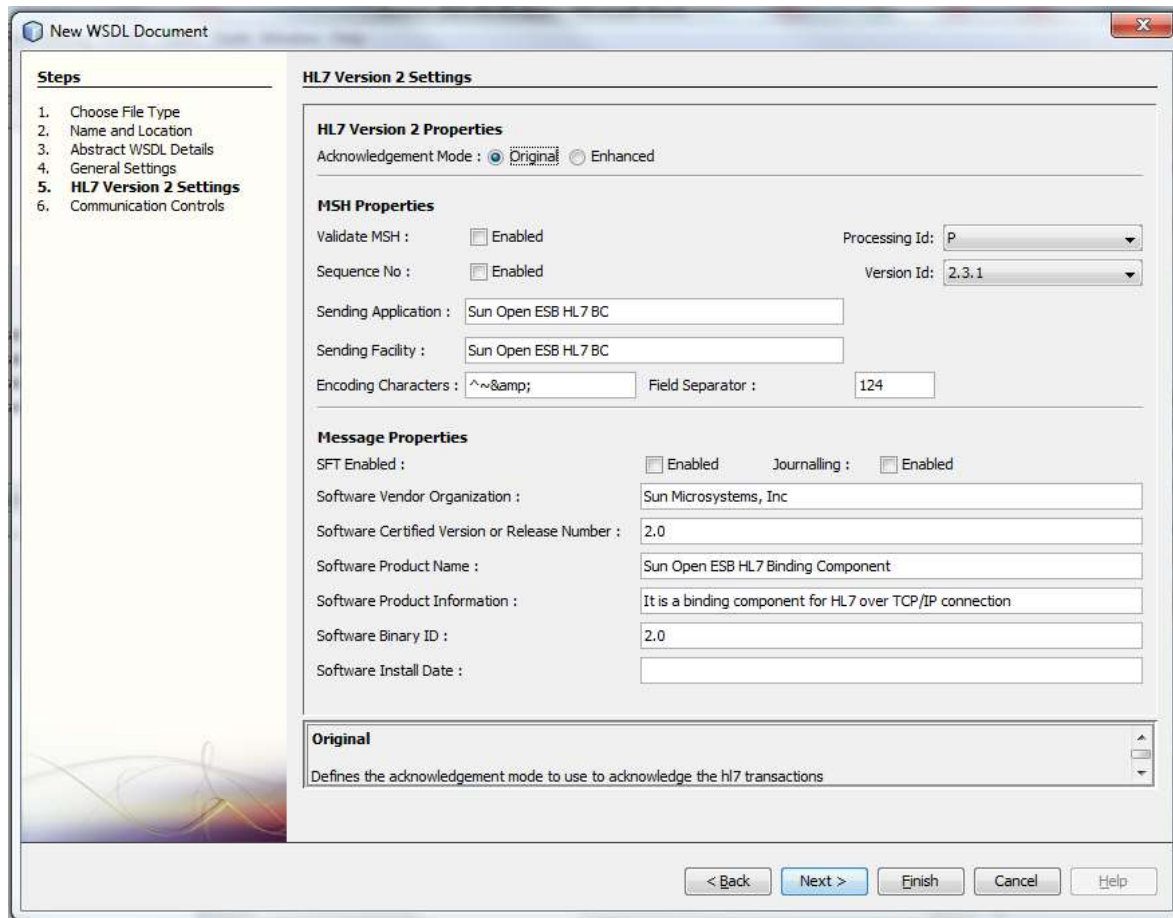


Figura D.22. Propiedades de la versión de HL7.

De esta forma tenemos el descriptor del servicio creado.

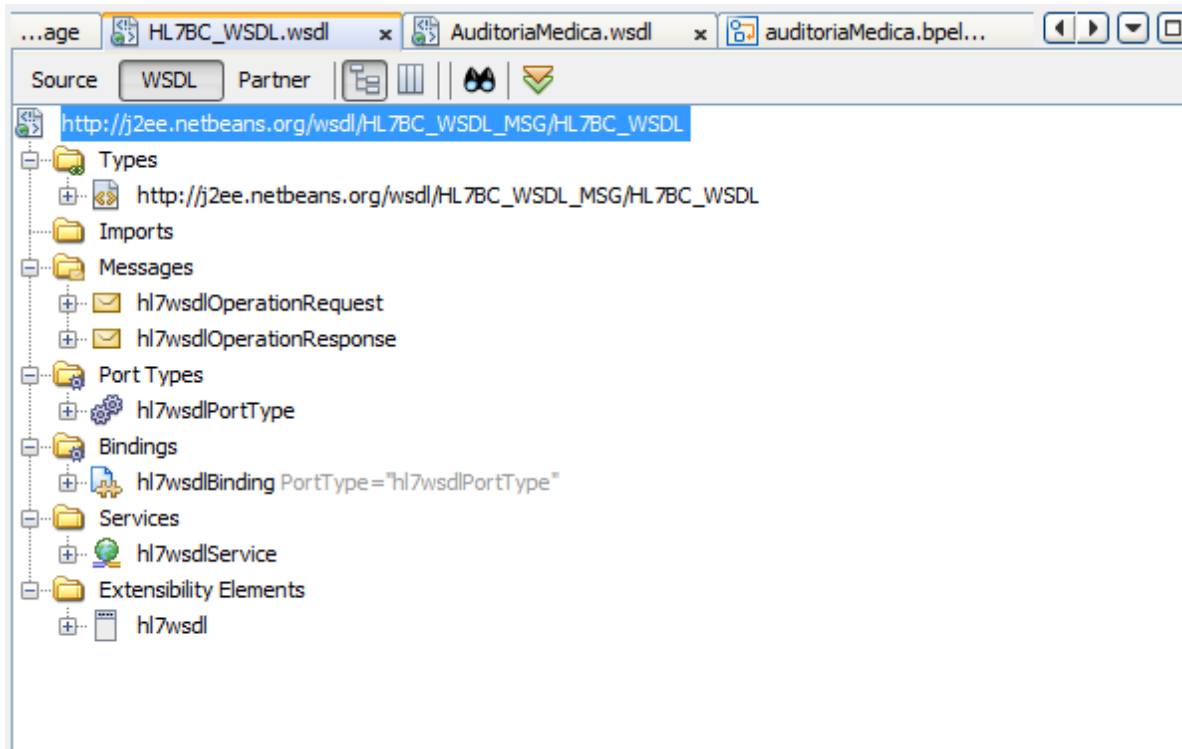


Figura D.23. WSDL creada.

Tipo de codificador empleado Figura D.23.

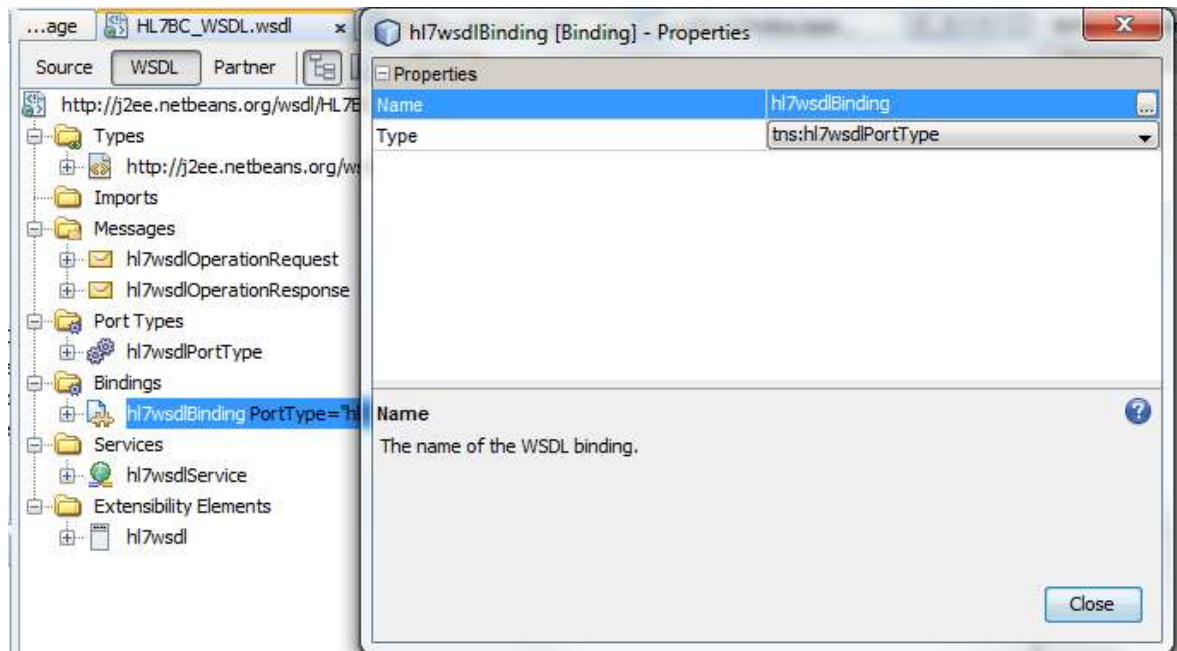


Figura D.24. WSDL Binding.



Ahora resta probar que todo este bien se da Validate XML al descriptor y deploy al proyecto.

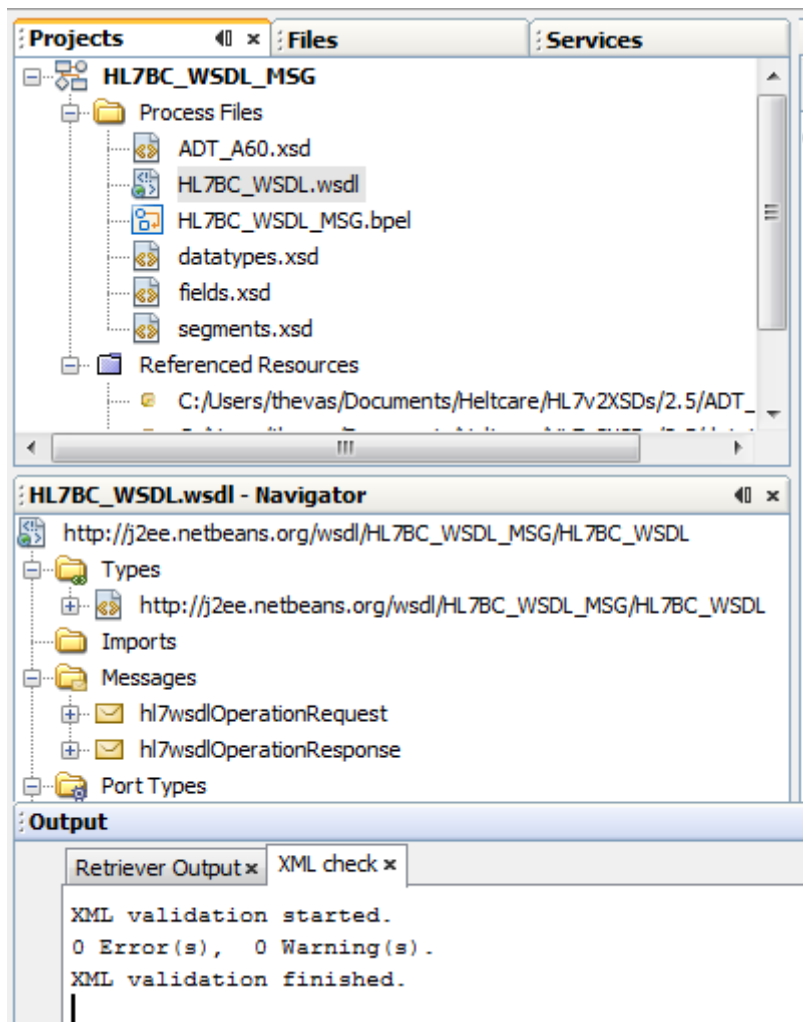


Figura D.25. Validar WSDL.

Creamos un proceso dando click bpel module y a este le agregamos un parnerLink.

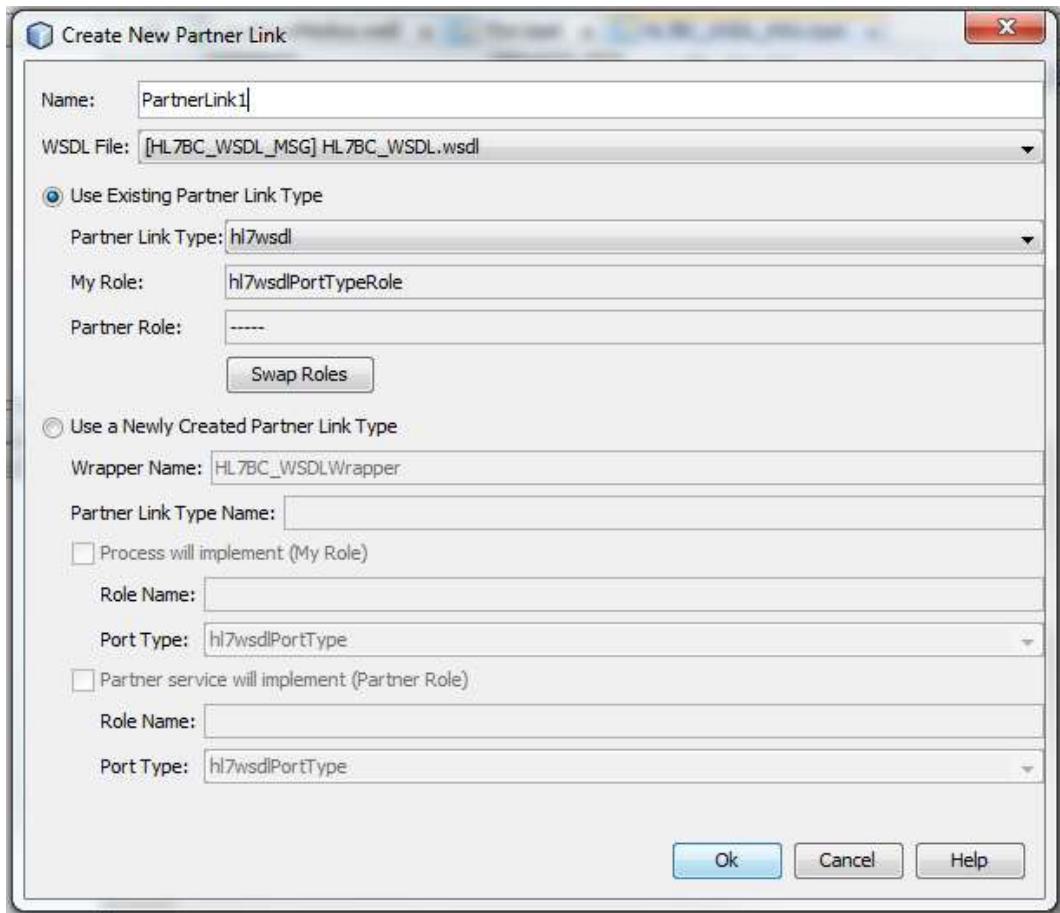


Figura D.26. ParnerLink del servicio.

Y nos crea el proceso Bpel con el que crearemos el

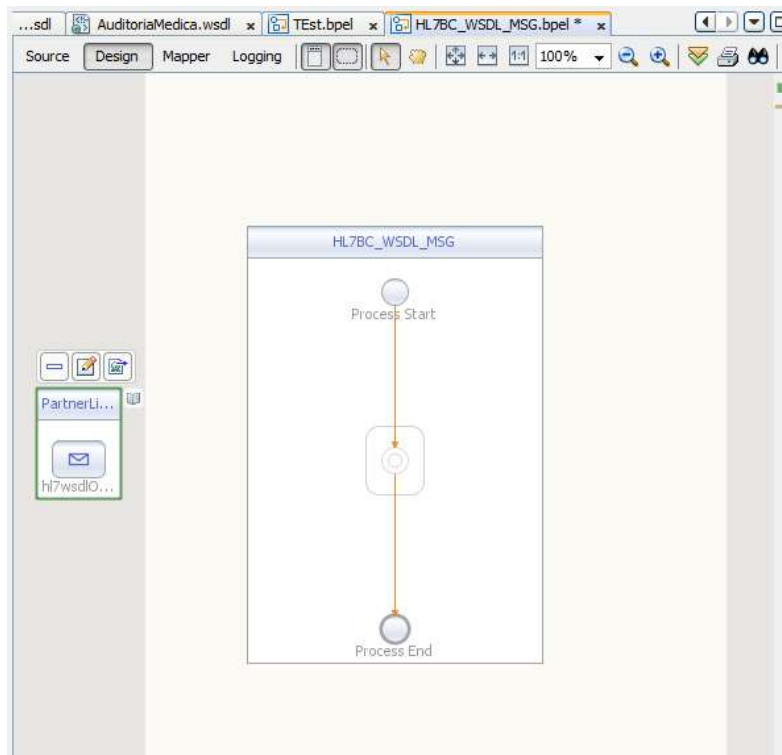


Figura D.27. Proceso vacío.

Hora creamos los compones de recibir el mensaje desde el parnerlink, no que se emplea la variable de entrada del servicio.

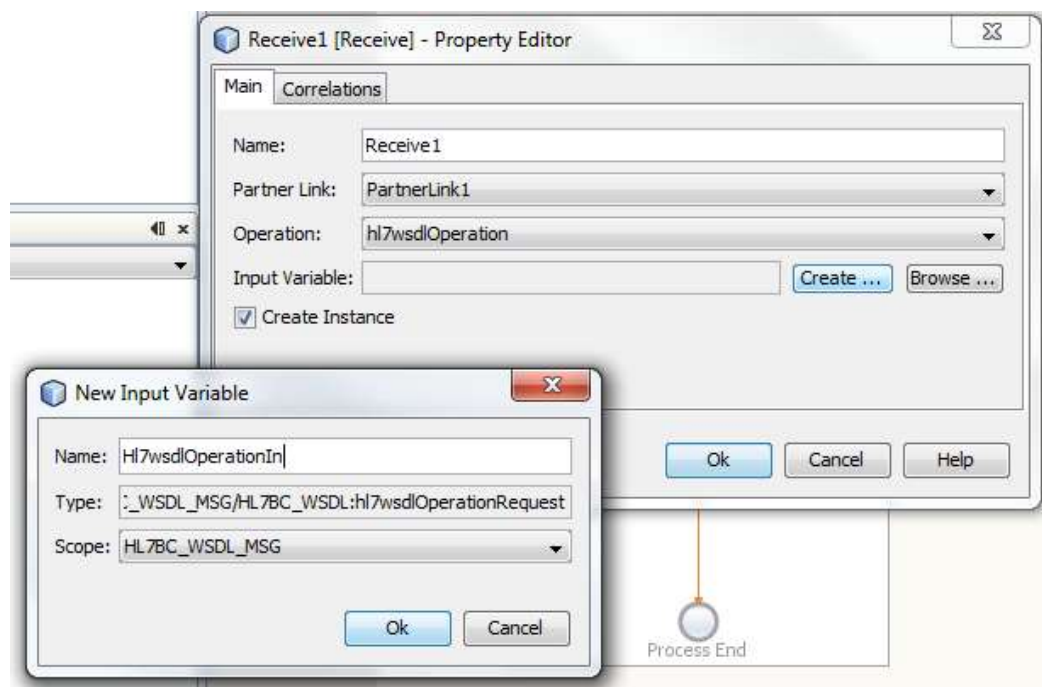


Figura D.28. Recibir Request.



Le agregamos una respuesta al proceso como muestra la Figura D.29.

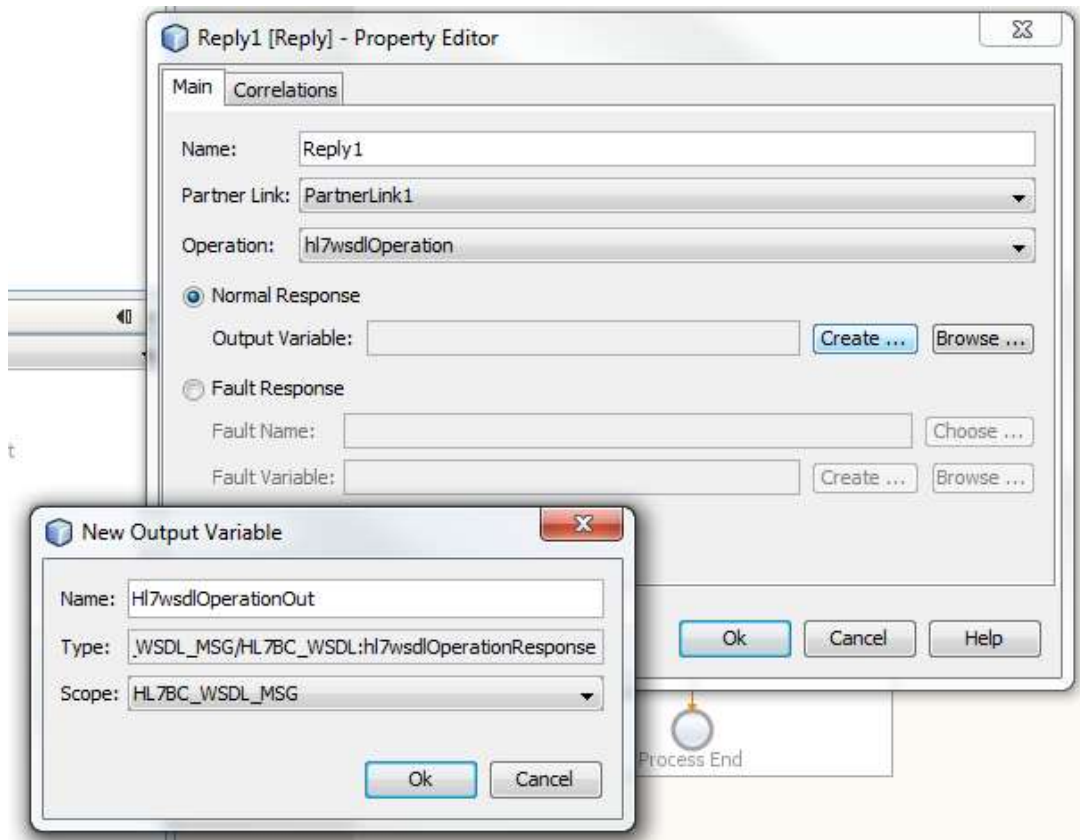


Figura D.29. Respuesta del Proceso.

De esta forma como muestran las siguientes imágenes creamos una asignación y una los respectivos asaing de las variables son seteados como muestra la Figura D.31.

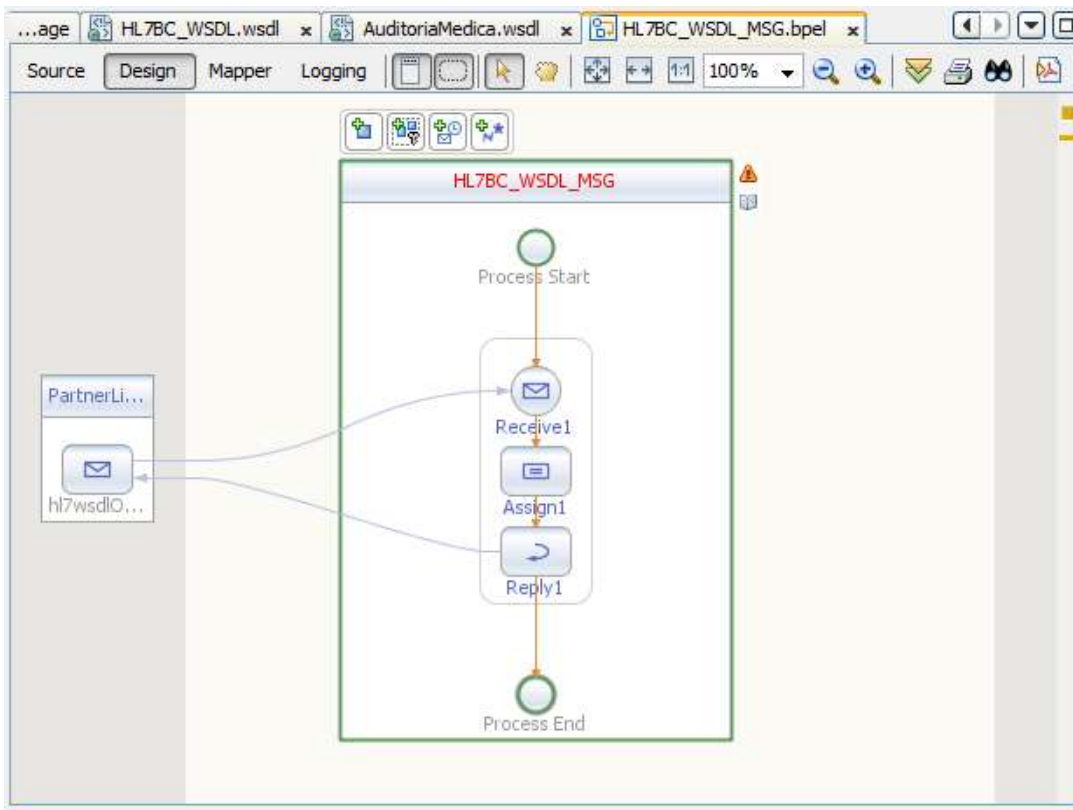


Figura D.30. agregamos una Asigancion.

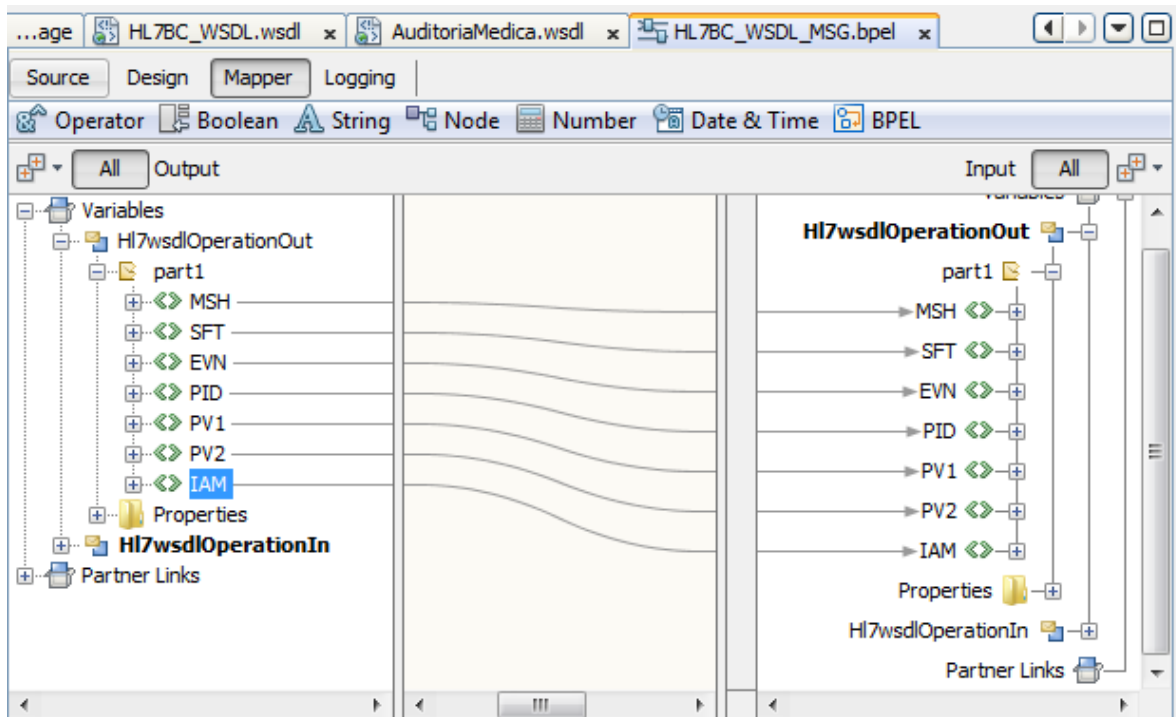


Figura D.31. Seteamos los valres de la asignacion que serán codificados por HL7.



D.4. PRUEBA DEL CODIFICADOR DESDE COMPONENTE APLICACIÓN.

Creamos una aplicación de prueba de como muestra la siguiente figura.

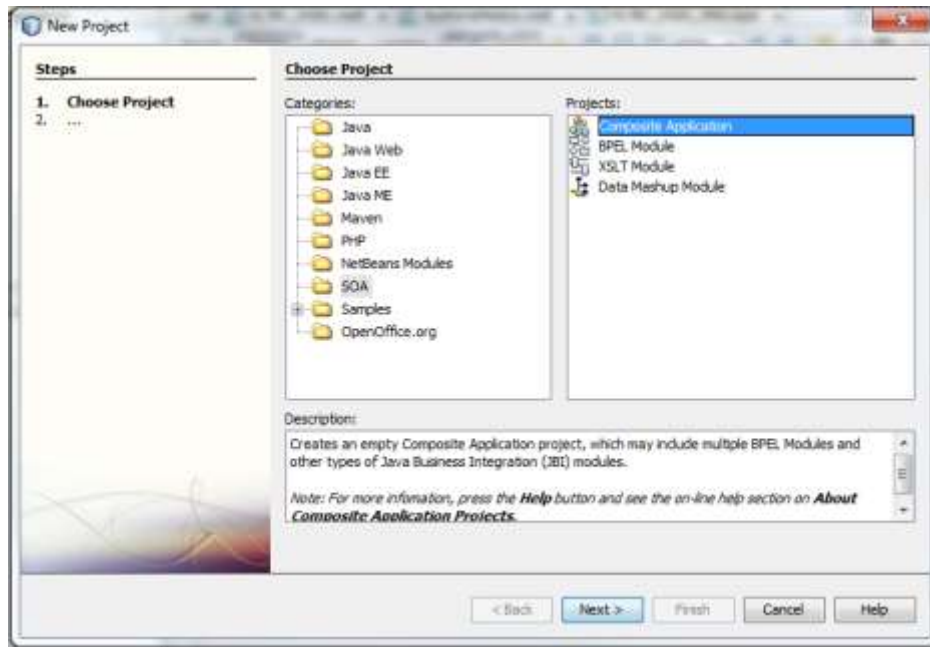


Figura D.32. Creamoa Aplicacion de composición.

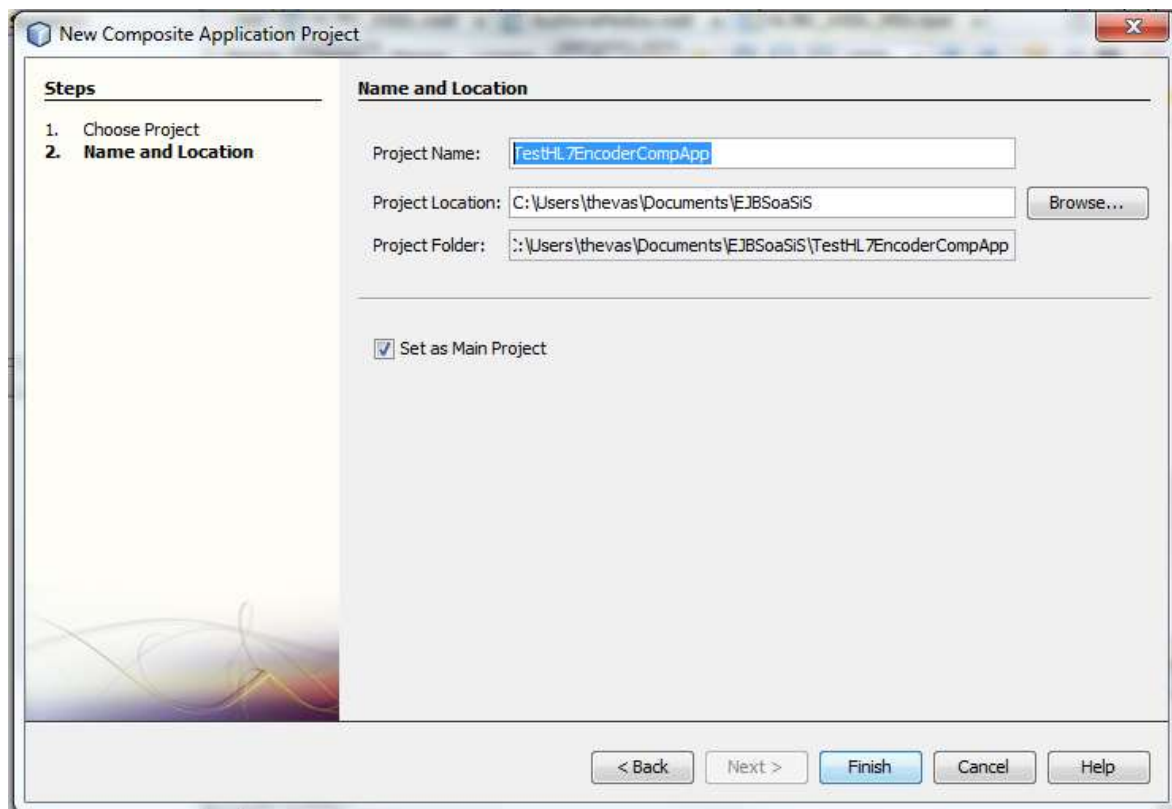


Figura D.33. Nombre de la Aplicacion.Figura D.34

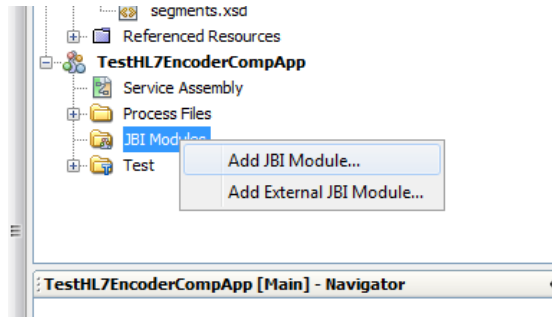


Figura D.35. Agregar Modulo JBI.

Como muestra la figura anterior agregamos el SWDL que creamos con soporte para HL7 Encoder.

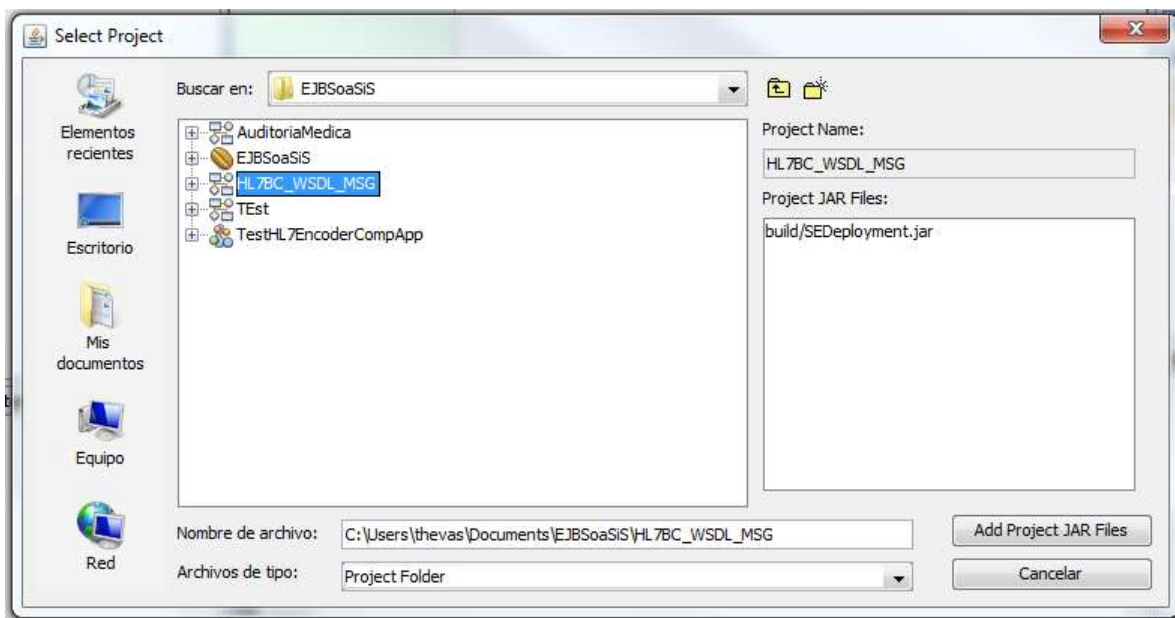
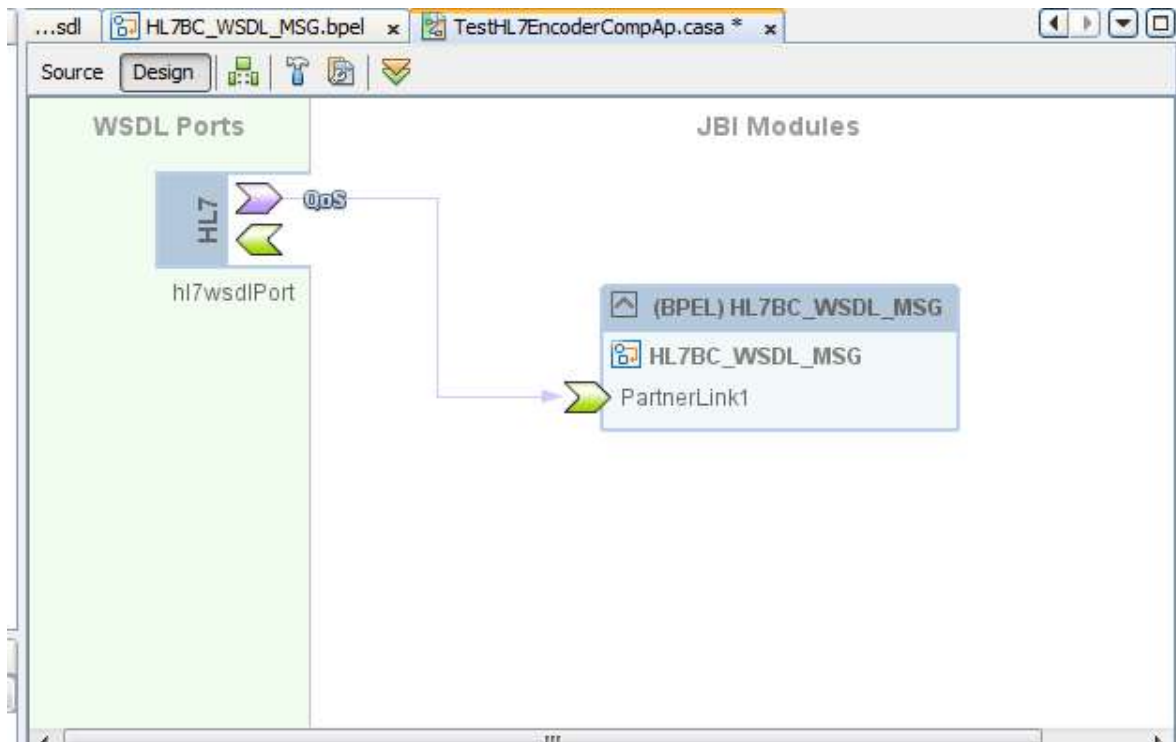


Figura D.36 Importar Servicio de codificador HL7.



A continuación vemos el

```
<?xml version="1.0" encoding="UTF-8"?>
<process
  name="HL7BC_WSDL_MSG"
  targetNamespace="http://enterprise.netbeans.org/bpel/HL7BC_WSDL_MSG/HL7BC_WSDL_MSG"
  xmlns:tns="http://enterprise.netbeans.org/bpel/HL7BC_WSDL_MSG/HL7BC_WSDL_MSG"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
  xmlns:sxt="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/Trace"
  xmlns:sxed="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/Editor"
  xmlns:sxeh="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/ErrorHandling"
  xmlns:sxed2="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/Editor2"      xmlns:ns0="urn:hl7-
org:v2xml">
  <import
    namespace="http://j2ee.netbeans.org/wsd/HL7BC_WSDL_MSG/HL7BC_WSDL"
    location="HL7BC_WSDL.wsdl" importType="http://schemas.xmlsoap.org/wsdl/">
  <partnerLinks>
    <partnerLink name="PartnerLink1" xmlns:tns="http://j2ee.netbeans.org/wsd/HL7BC_WSDL_MSG/HL7BC_WSDL"
    partnerLinkType="tns:hl7wsdl" myRole="hl7wsdlPortTypeRole"/>
  </partnerLinks>
  <variables>
    <variable
      name="hl7wsdlOperationOut"
      xmlns:tns="http://j2ee.netbeans.org/wsd/HL7BC_WSDL_MSG/HL7BC_WSDL"
      messageType="tns:hl7wsdlOperationResponse"/>
  </variables>
</process>
```



```
<variable name="HI7wsdlOperationIn"
xmlns:tns="http://j2ee.netbeans.org/wsdl/HL7BC_WSDL_MSG/HL7BC_WSDL"
messageType="tns:hl7wsdlOperationRequest"/>
</variables>
<sequence>
  <receive name="Receive1" createInstance="yes" partnerLink="PartnerLink1" operation="hl7wsdlOperation"
xmlns:tns="http://j2ee.netbeans.org/wsdl/HL7BC_WSDL_MSG/HL7BC_WSDL" portType="tns:hl7wsdlPortType"
variable="HI7wsdlOperationIn"/>
  <assign name="Assign1">
    <copy>
      <from>$HI7wsdlOperationOut.part1/ns0:MSH</from>
      <to>$HI7wsdlOperationOut.part1/ns0:MSH</to>
    </copy>
    <copy>
      <from>$HI7wsdlOperationOut.part1/ns0:SFT</from>
      <to>$HI7wsdlOperationOut.part1/ns0:SFT</to>
    </copy>
    <copy>
      <from>$HI7wsdlOperationOut.part1/ns0:EVN</from>
      <to>$HI7wsdlOperationOut.part1/ns0:EVN</to>
    </copy>
    <copy>
      <from>$HI7wsdlOperationOut.part1/ns0:PID</from>
      <to>$HI7wsdlOperationOut.part1/ns0:PID</to>
    </copy>
    <copy>
      <from>$HI7wsdlOperationOut.part1/ns0:PV1</from>
      <to>$HI7wsdlOperationOut.part1/ns0:PV1</to>
    </copy>
    <copy>
      <from>$HI7wsdlOperationOut.part1/ns0:PV2</from>
      <to>$HI7wsdlOperationOut.part1/ns0:PV2</to>
    </copy>
    <copy>
      <from>$HI7wsdlOperationOut.part1/ns0:IAM</from>
      <to>$HI7wsdlOperationOut.part1/ns0:IAM</to>
    </copy>
  </assign>
  <reply name="Reply1" partnerLink="PartnerLink1" operation="hl7wsdlOperation"
xmlns:tns="http://j2ee.netbeans.org/wsdl/HL7BC_WSDL_MSG/HL7BC_WSDL" portType="tns:hl7wsdlPortType"
variable="HI7wsdlOperationOut"/>
</sequence>
</process>
```

D. 4 CODIFICADOR GFORSE PARA MAPEAR HL7 v2 A HL7 v3

Como la compatibilidad que ofrece OpenESB para HL7 está basada en HL7 v2 las librerías de HL7 Gforse ofrece un conjunto de librerías para mapear mensajes del formato HL7 e versión a HL7 aversión 3, estas librerías esta basadas en HL7 V3 API.



<http://aurora.regenstrief.org/javasig/wiki/HL7v3Overview>

Las librerías se pueden obtener del siguiente enlace:

<http://gforge.hl7.org/gf/project/v2v3-mapping/frs/>

El uso de este plugin nos permite generar el componente JBI para hacer el mapeo según las reglas que definamos, se describirá el mapeo del XSD que se empleó en la prueba del servicio con HL7 para basarse la cabecera del mensaje HL7. Este componente se exportara al JBI container de OpenESB y luego este componente será consumiendo desde un EJB que es publicado como un servicio web.

Primero que todo obtener los plugins como se muestra en las próximas imágenes para el IDE Eclipse.

Eclipse Modeling Framework – SDK

XSD – XML Schemata Definition SDK

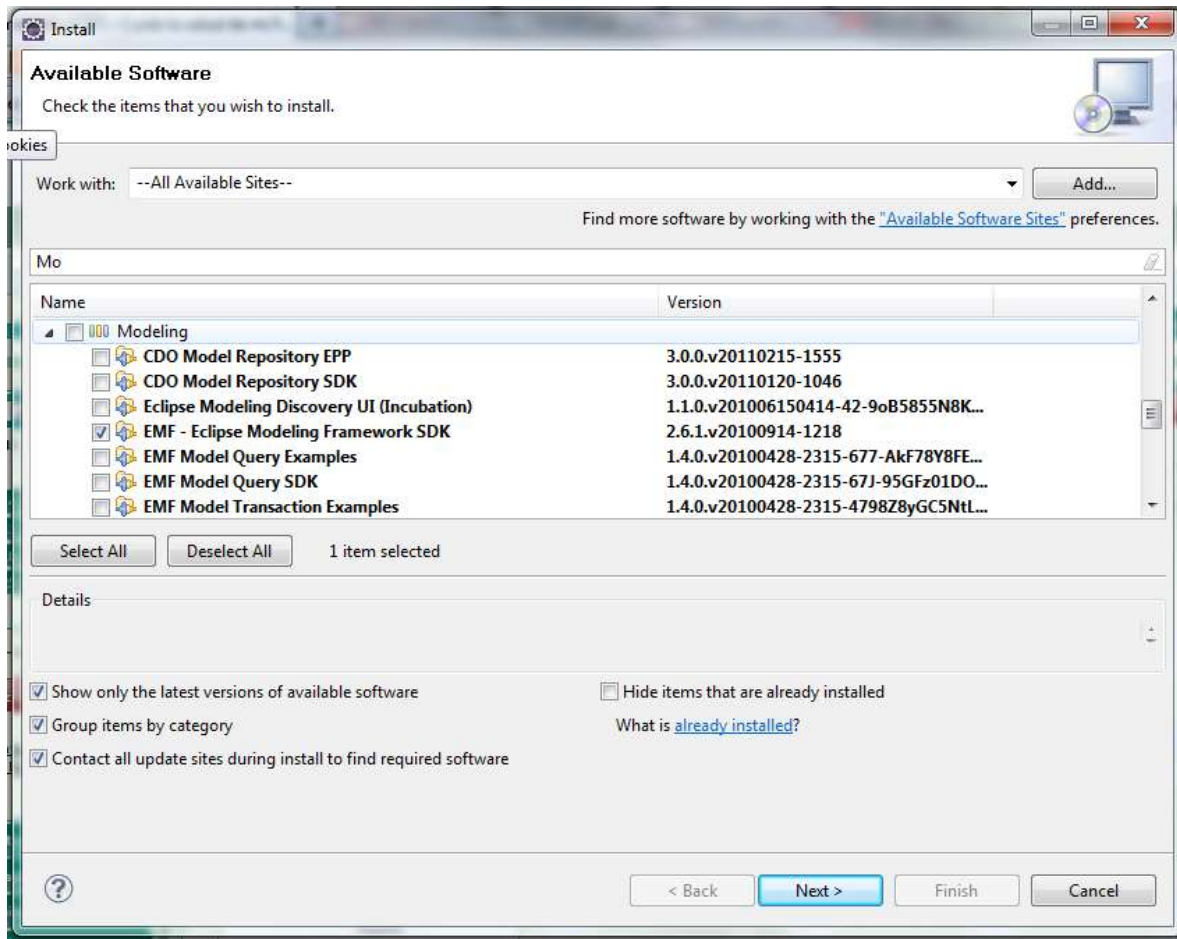


Figura D.37. Incluir soporte para Modelado.

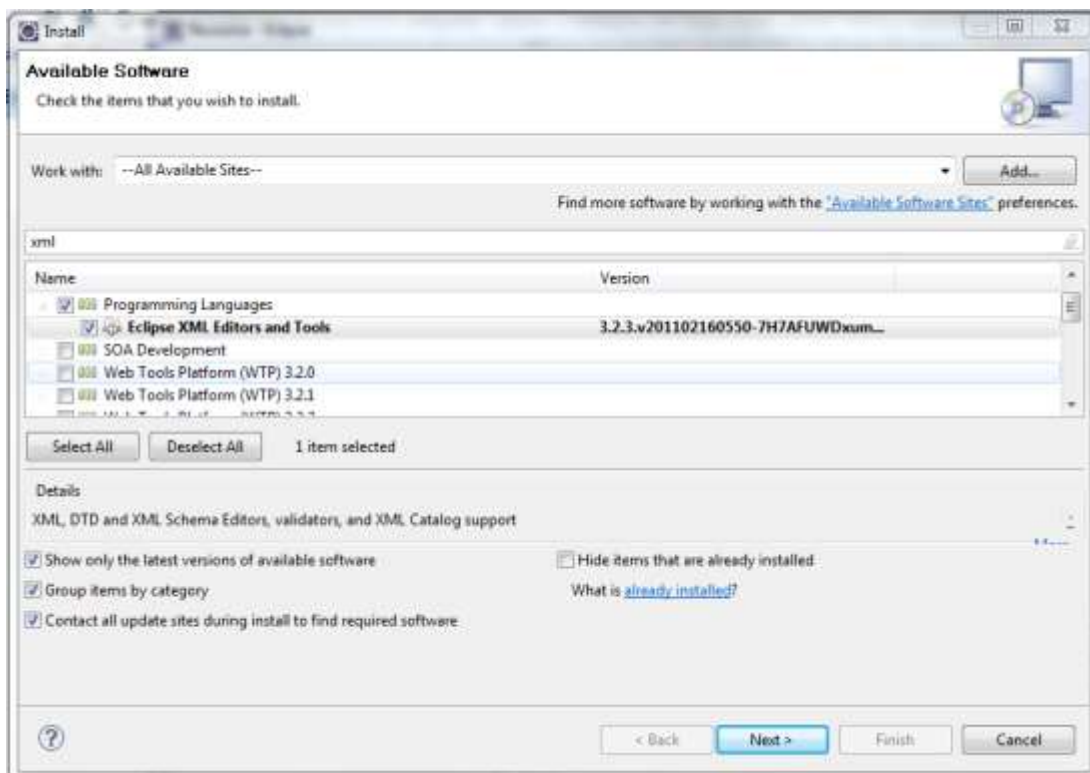
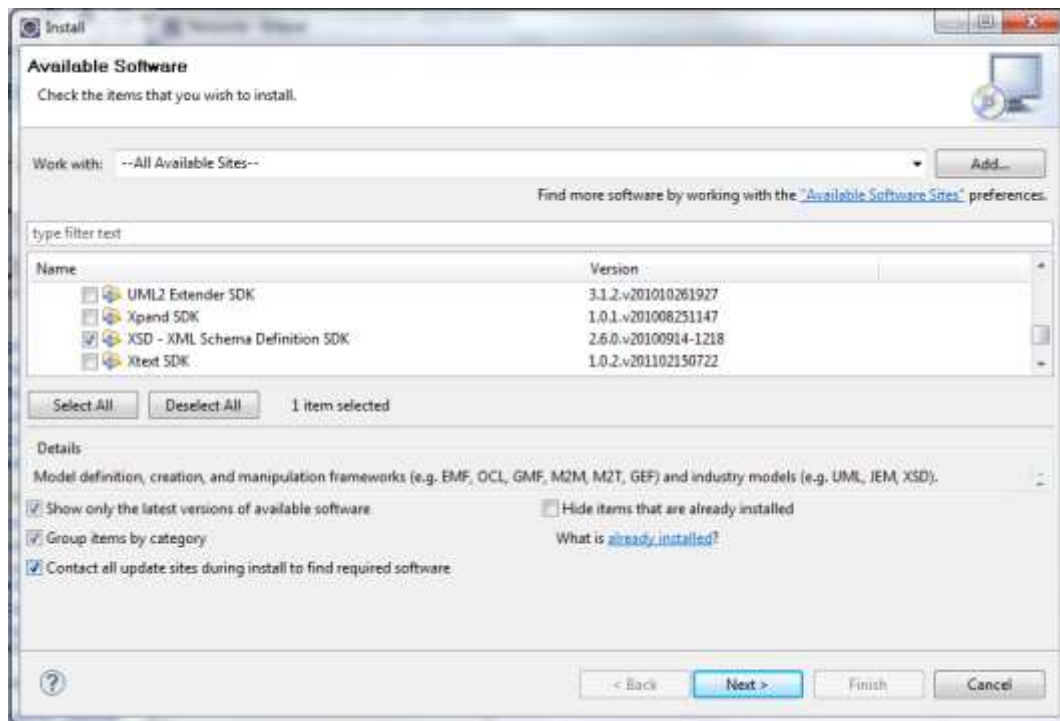


Figura D.38. Incluir soporte para definicion de XML.

Una vez instalados los complementos se debe descomprimir en el directorio donde está instalado eclipse el contenido de archivo descargado mapper_1.1.2.zip y reiniciamos el IDE.

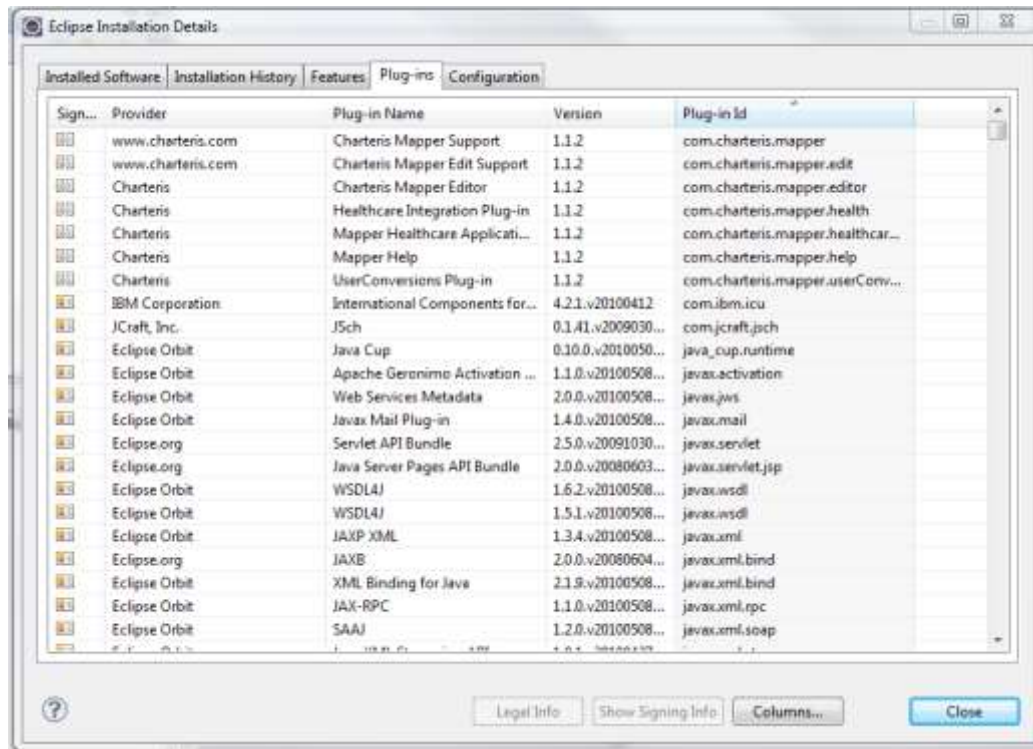


Figura D.39. Verificación del entorno.

Entrando al menú ayuda, luego about Eclipse y después Java EE IDE al ordenar descendientemente por Plugin ID debe presentar los plugins como en la imagen anterior.

En el directorio descomprimido se incluye con las herramientas asignador de dos visitas guiadas, una para las herramientas de asignación genérica (en un proyecto CarDemo) y otro para el HL7 V2 y V3 herramientas de mapeo (en un proyecto V3_2008NE).

Para iniciar una de las dos visitas guiadas, es necesario importar uno de los dos proyectos de demostración. Para ello, File> Import seguida de proyectos existentes en el área de trabajo (Existing Projects into Workspace). En el diálogo siguiente, seleccione "Mapper_Demo_Projects" la carpeta en el directorio raíz, a continuación, elija uno de los proyectos:

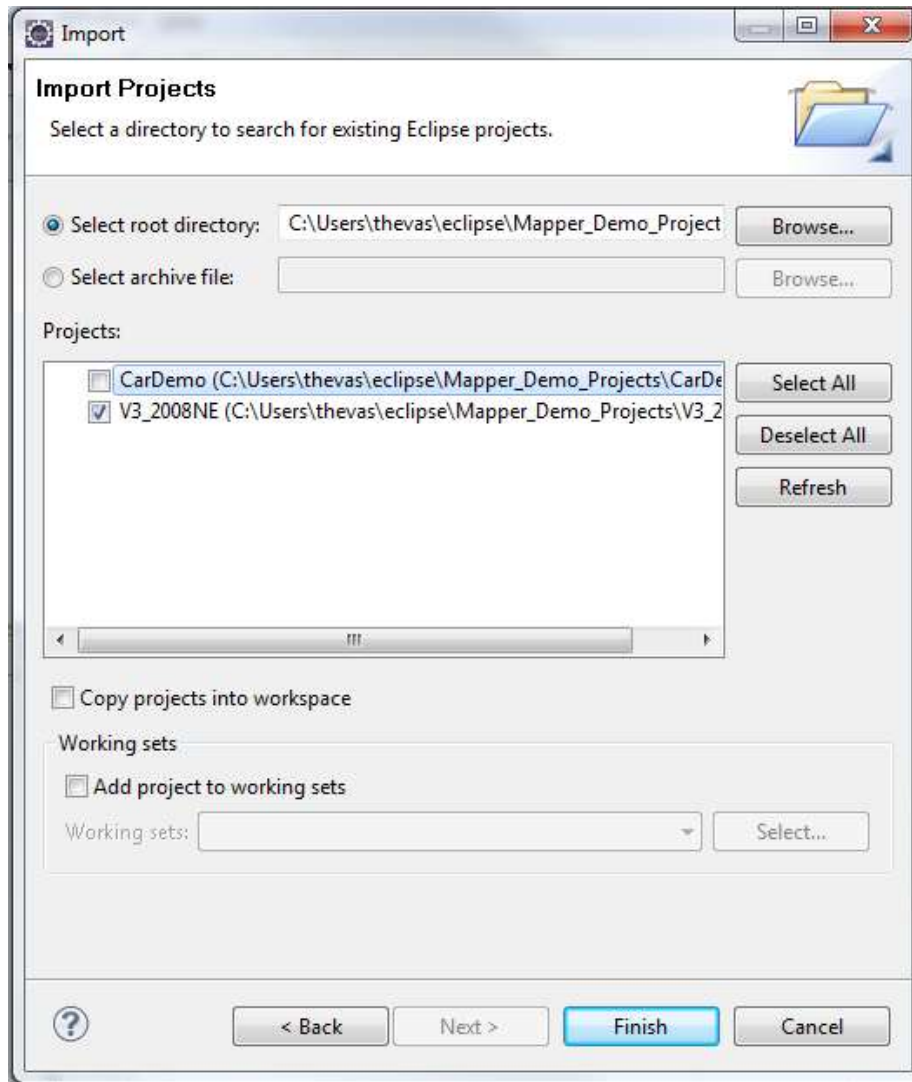


Figura D.40. Abrir proyecto plantilla.

Después de "terminar" el proyecto seleccionado deberá mostrar en 'Navigator' panel de la izquierda de Eclipse. Abra, abra 'MappingSets "la carpeta y haga clic en uno de los archivos del" asignador ". Ahora debería ver un mapa situado en el Editor de Mapper:

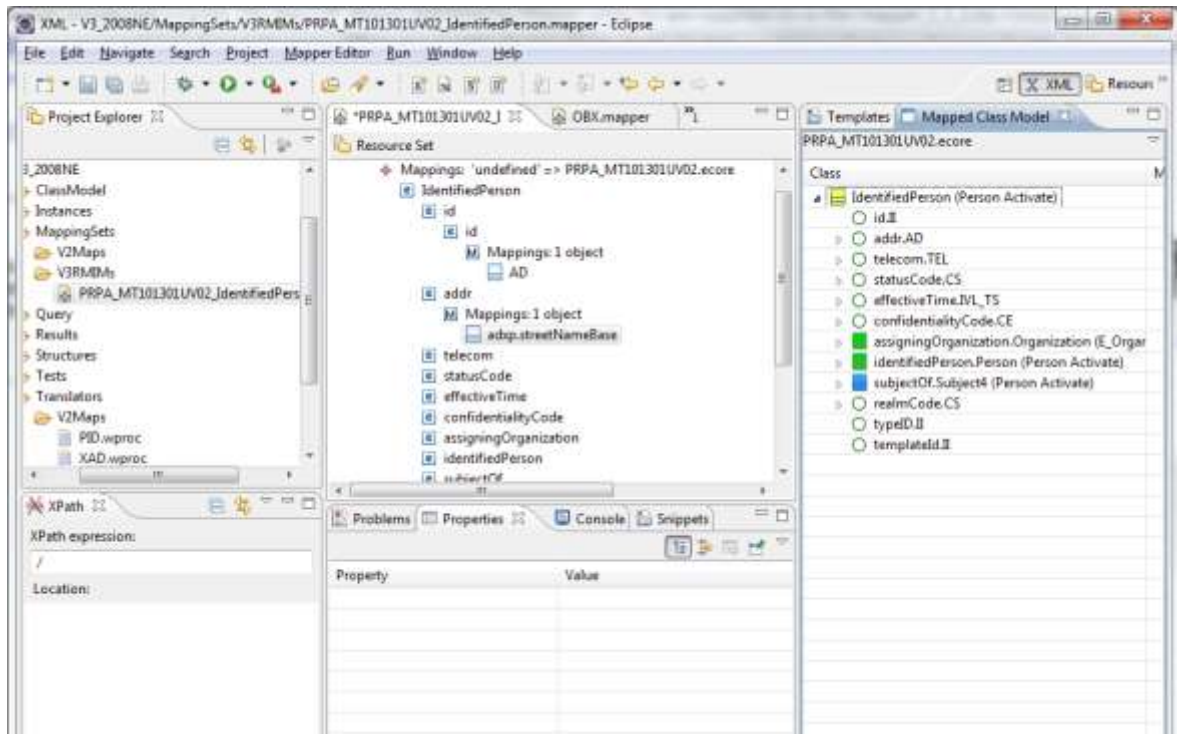


Figura D.41. Entorno de mapero HI7 v2 a v3.

Para la introducción más a las herramientas, siga las instrucciones de visita guiada para cualquier viaje. Estos se encuentran en la 'Introducción' sección de la Guía o asignador o asignador de Salud Guía de aplicaciones, en Ayuda Eclipse.

En el Mapped Class Model esta al mensaje genérico HL7 v2, se da click en la propiedad a mapear y se sitúa luego en el PRPA_MTT013xxx se da click derecho ya Add Map to Event.

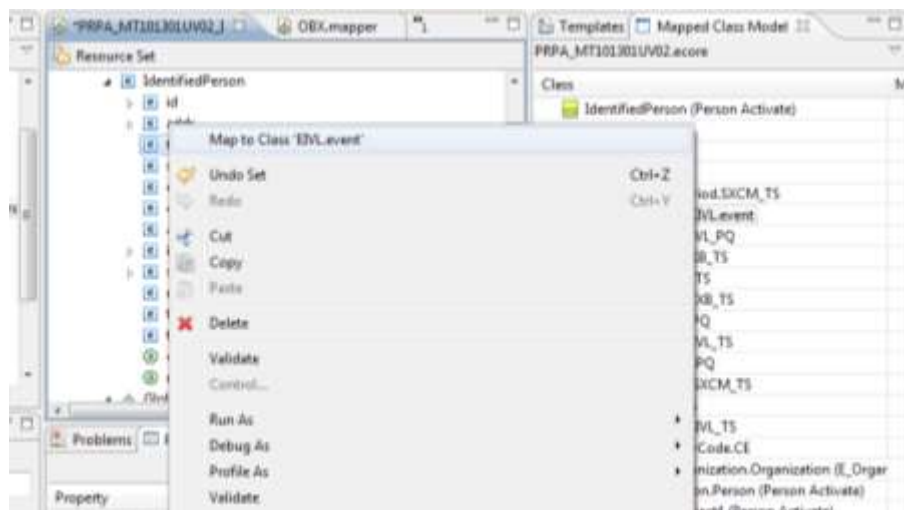


Figura D.42. Mapeo de un atributo.

Al finalizar se guarda se guarda y se exporta como un JAR file.



D.5 CONSTRUCCIÓN DE UN SERVICIO WEB DESDE UN EJB PARA HABILITAR LA TRASFORMACIÓN.

El archivo obtenido del mapeo de la sección anterior se debe configurar de manera similar a como se realizó en la parte D.1.1.

El proceso de creación de un servicio web a partir de un EJB se detalla en el Anexo F sección F.3.

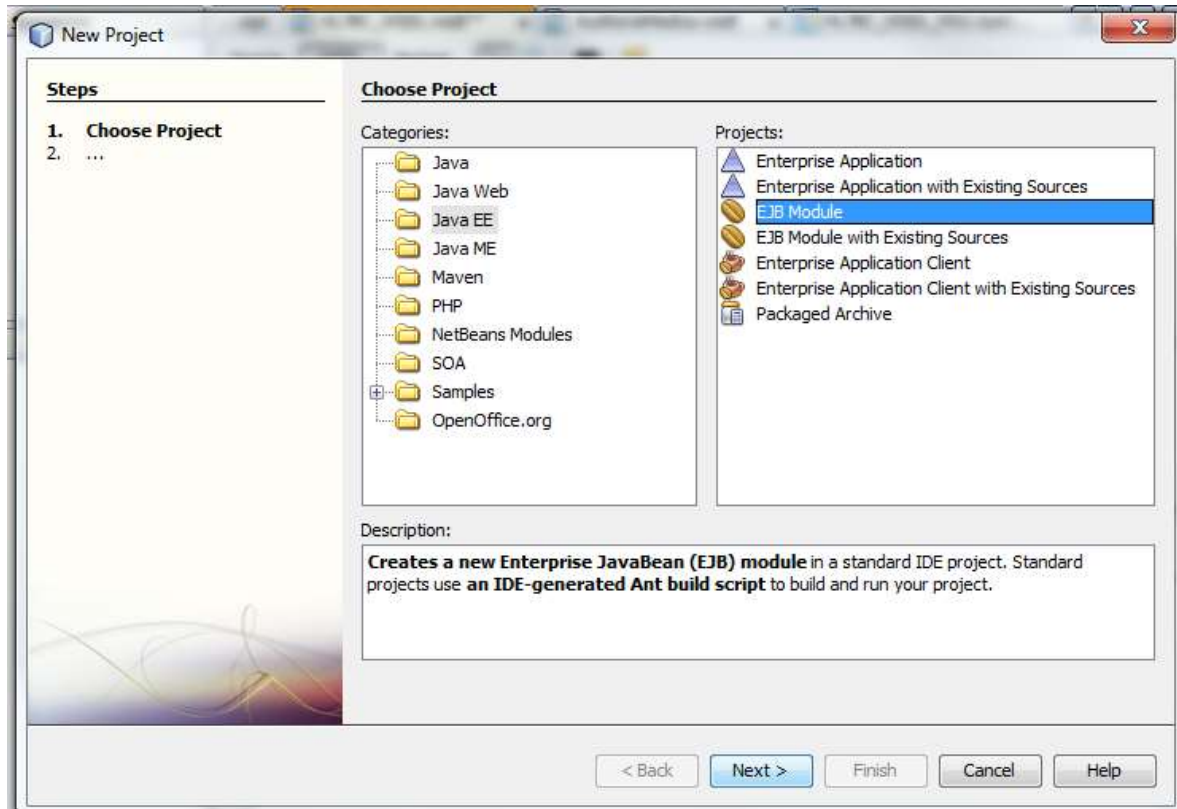
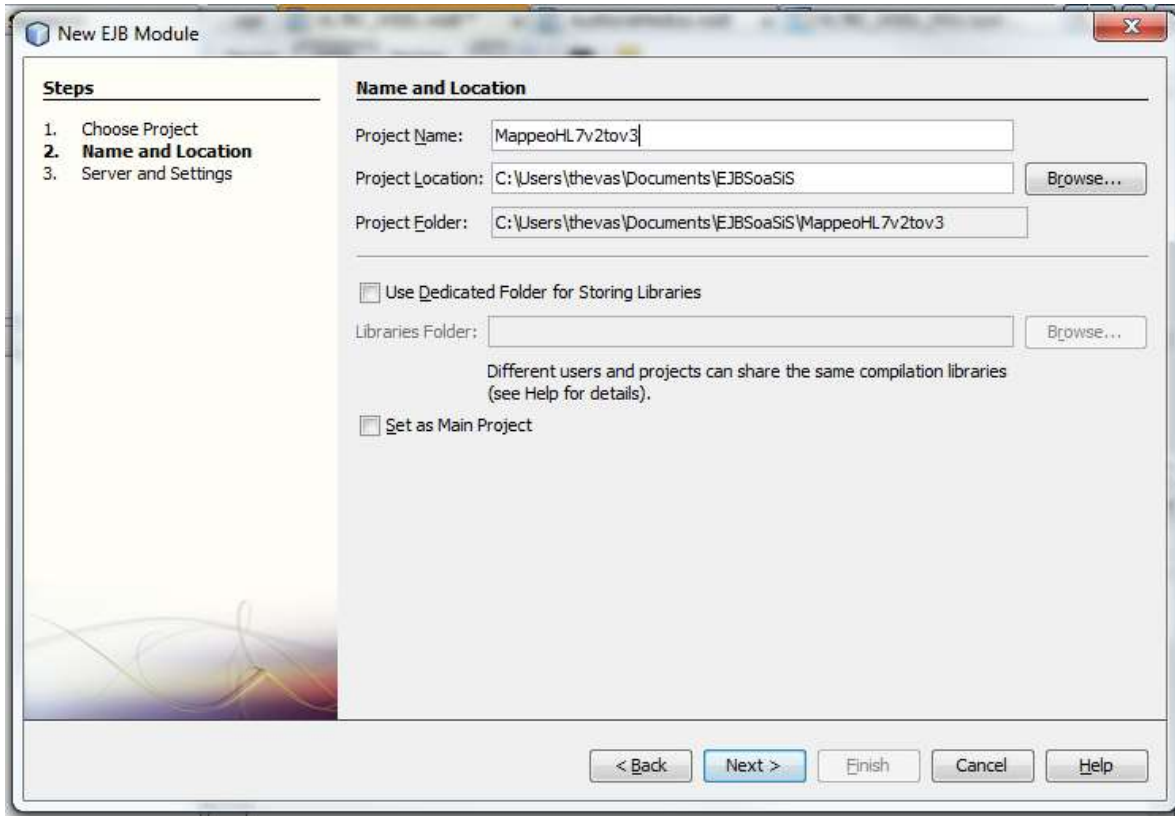


Figura D.43. Creacion de un servicio para habilitar la trasformacion.





E PROTOCOLO HL7 EN LA ARQUITECTURA ORIENTADA A SERVICIOS

Para efectos de validación del uso del estándar de interoperabilidad semántica (HL7) en la arquitectura de implementación, se agrega en el servicio de aplicación consulta de epicrisis un campo de respuesta tipo cadena con el mensaje y la información con el estándar HL7. Para lograr esto se habilitó el Business Component de HL7 versión 3 como se describe en detalle en el anexo D (sección D.2, Instalación de Building Component).

Para definir la estructura del mensaje se empleó la guía de implementación para resultados de laboratorio basada en el estándar HL7 para Arquitectura del Documento Clínico (Clinical Document Architecture, CDA) versión v21, propuesta por HL7 Colombia. Basados en esta especificación se construye el mensaje como se detalla en el anexo D. A continuación se describe el encabezado del CDA:

```
<?xml version="1.0"?>
<ClinicalDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" moodCode="EVN"
xmlns="urn:hl7-org:v3 CDA.xsd">
  <typeId root="2.16.840.1.113883.1.3" extension="POCD_HD000040" />
  <id root="2.16.840.1.113883.19.4.x" extension="c266" />
  <code code="26436-6" codeSystem="2.16.840.1.113883.6.1"
codeSystemName="LOINC" displayName="Epicrisis" />
  <title>Epicrisis</title>
  <effectiveTime value="20081209001500" />
  <confidentialityCode code="R" codeSystem="2.16.840.1.113883.5.25" />
  <languageCode codeSystem="2.16.840.1.113883.6.121" />
  <recordTarget typeCode="RCT" contextControlCode="OP">
    <patientRole classCode="PAT">
      <id root="2.16.840.1.113883.19.5" extension="55140250" />
      <addr use="HP">
        <streetAddressLine>Carrera 29A1 No. 12B-105</streetAddressLine>
      </addr>
      <patient classCode="PSN" determinerCode="INSTANCE">
        <id root="2.16.840.1.113883.19.5" extension="55140250" />
        <name>
          <given>JUAN</given>
          <given>DANIEL</given>
          <family>MORALES</family>
          <family>GUTIERREZ</family>
        </name>
        <administrativeGenderCode code="M"
codeSystem="2.16.840.1.113883.5.1" />
        <birthTime value="20090203" />
      </patient>
      <providerOrganization classCode="ORG" determinerCode="INSTANCE">
        <id root="2.16.840.1.113883.19.5" extension="890.307.200-5" />
        <name>
          <suffix qualifier="LS">S.A.</suffix>Hospital Susana Lopez de Valencia
        </name>
        <telecom value="fax: 57 8381151" />
        <telecom value="tel: 57 8211721" />
        <telecom value="tel: 57 8382411" />
        <telecom value="mailto: informacion@hosusana.gov.co" />
        <telecom value="http: http://www.hosusana.gov.co " />
        <addr>
          <streetAddressLine>Calle 15 N° 17A - 196 La Ladera</streetAddressLine>
          <city>Popayán</city>
          <state>Cauca</state>
          <country>Colombia</country>
        </addr>
      </providerOrganization>
    </patientRole>
```



```
</recordTarget>
<author typeCode="AUT" contextControlCode="OP">
  <time value="20090203001746" />
  <assignedAuthor classCode="ASSIGNED">
    <id root="2.16.840.1.113883.19.5" extension="KP00017" />
    <assignedAuthoringDevice determinerCode="INSTANCE">
      <code code="dev1" />
      <manufacturerModelName>Peter Pinchao</manufacturerModelName>
      <softwareName>SOA SIS</softwareName>
    </assignedAuthoringDevice>
    <representedOrganization classCode="ORG" determinerCode="INSTANCE">
      <id root="2.16.840.1.113883.19.5" extension="890.307.200-5" />
    </representedOrganization>
  </assignedAuthor>
</author>

<componen>

  </componen>
</ClinicalDocument>
```



F ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LOS APLICATIVOS AUDITORIA Y TELE CONSULTA.

En este anexo se presenta el desarrollo de los aplicativos de conforman los procesos de negocio estudiados, aunque se emplea un enfoque metodológico totalmente orientado a servicios, e análisis de requerimientos y diseño se puede emplear bajo la metodología RUP como se presenta en es este aparte.

F.1. ANÁLISIS.

Se describen a continuación los problemas que abordan cada uno de los subproductos del sistema solución.

F.1.1. DESCRIPCIÓN DEL PROBLEMA.

Proceso de negocio Tele-consulta.

Se requiere implementar un sistema de tele-consulta que brinde al personal médico de la institución y de otras instituciones suscritas a la Red Nacional Académica de tecnología Avanzada (RENATA), para hacer consulta de la epicrisis de un paciente, en caso de ser necesario brindar la posibilidad de acceder a la historia clínica, a imágenes diagnósticas y resultados de exámenes de laboratorio.

Proceso de negocio Auditoria Médica.

Se requiere brindar a los auditores médicos de las diferentes entidades administradoras de salud, la un sistema informático que les permita confrontar los datos de una factura con los datos de prestación de servicio relacionados con dicha factura, para lo cual debe poder marcar la aprobación o desaprobación de la misma, en tal caso debe poder crear, borrar y editar observaciones al respecto de los servicios facturado que motivan el hecho de desaprobación.

F.1.2. DIAGRAMA DE CASOS DE USO.

Cosos de Uso del sistema Tele-consulta.

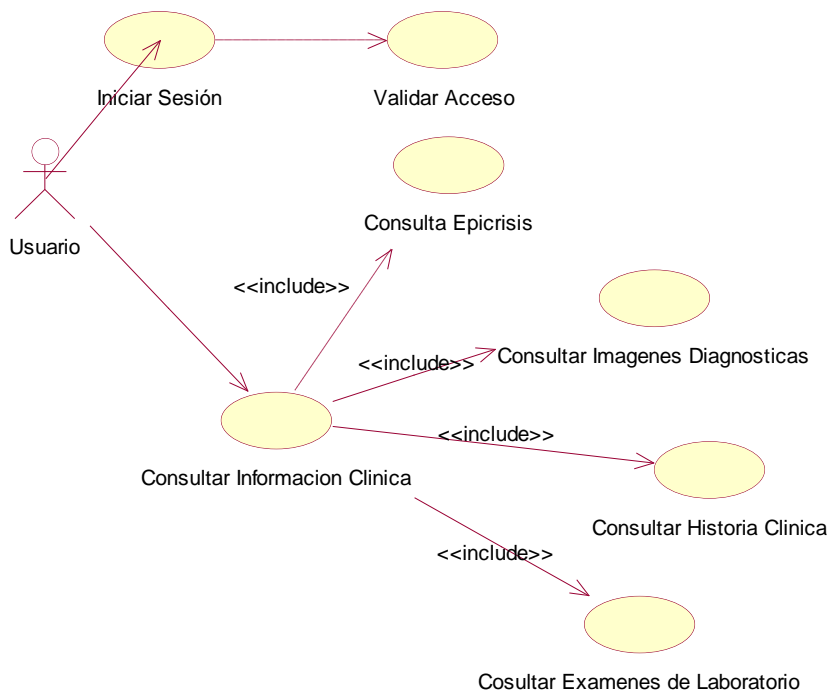


Figura F.1. Casos de uso sistema de Tele-consulta.

Se describen a continuación los casos de Uso.

Tabla 13. Caso de Uso Iniciar Sesión.

| | |
|---|--|
| Caso de uso: | Iniciar Sesión |
| Autores | Peter Emerson Pinchao. |
| Actores | |
| <ul style="list-style-type: none"> • Usuario • Servicios Web. | |
| Pre-Condición: | Ninguna |
| Descripción: | Se inicia cuando el actor desea acceder al sistema de mediante proveyendo los datos necesarios para tal fin. |
| CURSO NORMAL DE LOS EVENTOS | |
| Acción del Actor | Respuesta del Sistema |
| 1. El <actor> indica al sistema que requiere un iniciar a los servicios de tele consulta. | 2. El sistema le solicita los datos correspondientes nombre de usuario y contraseña. |
| 3. El <actor> ingresa los datos solicitados por el sistema | 4. El sistema pasa los datos a sistema de validación. |
| Pos-Condiciones | El usuario debe estar registrado en base de datos o en el directorio de autenticación. |



Tabla 14. Caso de Uso Validar Acceso.

| | |
|---|---|
| Caso de uso: | Validar Acceso |
| Autores | Peter Emerson Pinchao. |
| Actores | |
| <ul style="list-style-type: none"> • Usuario • Servicios Web. | |
| Pre-Condición: | Ninguna |
| Descripción: | Se inicia cuando el Caso de Uso Iniciar Sesión solicita el acceso los recursos del sistema. |
| CURSO NORMAL DE LOS EVENTOS | |
| Acción del Actor | Respuesta del Sistema |
| | 1. El sistema da acceso a los recursos del sistema.. |
| 2. El <actor> puede visualizar los recursos del sistema. | |
| Pos-Condiciones | Ninguna. |

Tabla 15. Caso de Uso Consultar Información Clínica.

| | |
|--|---|
| Caso de uso: | Consultar Información Clínica. |
| Autores | Peter Emerson Pinchao. |
| Actores | |
| <ul style="list-style-type: none"> • Usuario | |
| Pre-Condición: | Se requiere haber validado el acceso al sistema. |
| Descripción: | Este caso de uso es un conjunto composición de otros casos de uso. |
| CURSO NORMAL DE LOS EVENTOS | |
| Acción del Actor | Respuesta del Sistema |
| | 1. el sistema muestra los recursos disponibles para poder acceder a la información clínica disponible |
| 2. el <actor> puede seleccionar los criterios a emplear para hacerse a la información de su interés. | |
| Pos-Condiciones | El <actor> debe estar seguro del ingreso del paciente de su interés. |

Tabla 16. Caso de uso Consultar Epicrisis.

| | |
|---|---|
| Caso de uso: | Consultar Epicrisis. |
| Autores | Peter Emerson Pinchao. |
| Actores | |
| <ul style="list-style-type: none"> • Usuario • Servicios Web. | |
| Pre-Condición: | El <actor> debe estar seguro del ingreso del paciente de su interés. |
| Descripción: | Este caso de uso inicia cuando el <actor> desea consultar la información de la epicrisis de un usuario de la IPS. |
| CURSO NORMAL DE LOS EVENTOS | |
| Acción del Actor | Respuesta del Sistema |



| | |
|--|---|
| Caso de uso: | Consultar Epicrisis. |
| 1. el <actor> envía la solicitud de consulta de epicrisis, con los datos que necesarios para identificar un equívocamente. | |
| | 2. El sistema le muestra las epicrisis existentes del paciente ordenadas en forma temporal descendientemente. |
| 3. El <actor> selecciona la epicrisis que desea consultar de acuerdo al ingreso que sea de su interés. | |
| | 4. El sistema consulta y muestra los datos de la epicrisis. |
| Pos-Condiciones | Ninguna. |

Tabla 17. Caso de Uso Consultar Historia Clínica

| | |
|--|--|
| Caso de uso: | Consultar Historia Clínica. |
| Autores | Peter Emerson Pinchao. |
| Actores | |
| <ul style="list-style-type: none"> • Usuario • Servicios Web. | |
| Pre-Condición: | Haber consultado un epicrisis. |
| Descripción: | Inicia cuando el <actor> desea consultar los detalles clínicos del paciente. |
| CURSO NORMAL DE LOS EVENTOS | |
| Acción del Actor | Respuesta del Sistema |
| | 1. El sistema muestra la Historia clínica epicrisis |
| 2. El <actor> requiere los detalles de la epicrisis para ello solicita los datos de la historia clínica. | . |
| | 3. El Sistema presenta los datos de la historia clínica. |
| Pos-Condiciones | Ninguna. |

Tabla 18. Caso de Uso Consultar Exámenes de Laboratorio.

| | |
|---|--|
| Caso de uso: | Consultar Exámenes de Laboratorio |
| Autores | Peter Emerson Pinchao. |
| Actores | |
| <ul style="list-style-type: none"> • Usuario • Servicios Web. | |
| Pre-Condición: | Haber consultado un epicrisis. |
| Descripción: | Inicia cuando el <actor> desea consultar los detalles clínicos del paciente. |
| CURSO NORMAL DE LOS EVENTOS | |
| Acción del Actor | Respuesta del Sistema |
| | 1. El sistema muestra la Historia clínica epicrisis |



| | |
|---|---|
| Caso de uso: | Consultar Exámenes de Laboratorio |
| 2. El <actor> requiere los detalles de la epicrisis para ello solicita los resultados de exámenes clínicos. | . |
| | 3. El Sistema presenta los datos de los exámenes clínicos y sus resultados. |
| Pos-Condiciones | Ninguna. |

Tabla 19. Consultar Imágenes Diagnosticas.

| | |
|---|--|
| Caso de uso: | Consultar Imágenes Diagnosticas. |
| Autores | Peter Emerson Pinchao. |
| Actores | |
| <ul style="list-style-type: none"> • Usuario • Servicios Web. | |
| Pre-Condición: | Haber consultado un epicrisis. |
| Descripción: | Inicia cuando el <actor> desea consultar los detalles clínicos del paciente. |
| CURSO NORMAL DE LOS EVENTOS | |
| Acción del Actor | Respuesta del Sistema |
| | 1. El sistema muestra la Historia clínica epicrisis |
| 2. El <actor> requiere los detalles de la epicrisis para ello solicita las imágenes diagnósticas. | . |
| | 3. El Sistema presenta las imágenes diagnósticas. |
| Pos-Condiciones | Ninguna. |

Casos de Uso del sistema de Auditoria Médica.

En este punto se describen presentas los casos de uso del sistema de auditoria médica y se describen, sin considerar la descripción de aquellos que se encuentran entre los casos de uso de del sistema tele-consulta.

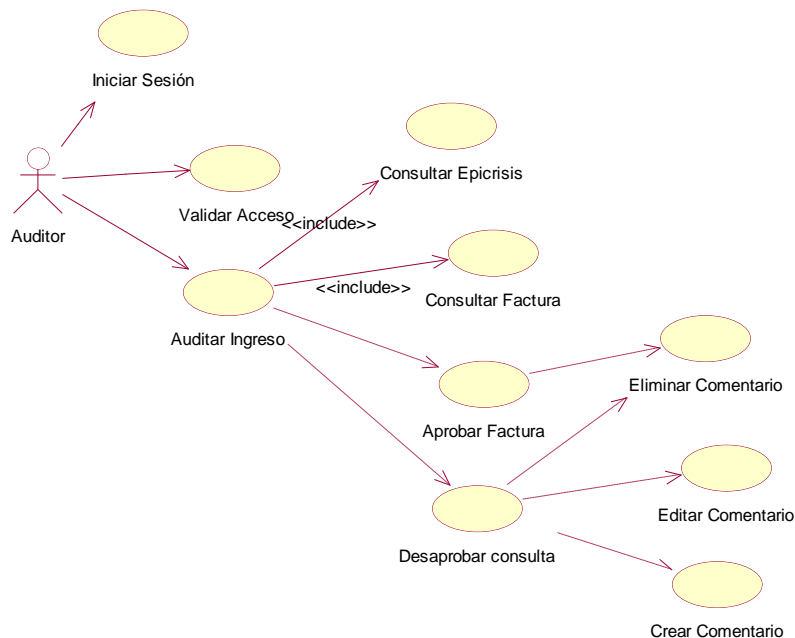


Figura F.2. Casos de uso Autoria Medica.

F.1.2.1. Descripción de casos de uso.

Tabla 20. Caso de Uso Auditar Ingreso.

| | |
|---|---|
| Caso de uso: | Consultar Auditar Ingreso |
| Autores | Peter Emerson Pinchao. |
| Actores | |
| <ul style="list-style-type: none"> • Usuario | |
| Pre-Condición: | Se requiere haber validado el acceso al sistema. |
| Descripción: | Este caso de uso es un conjunto composición de otros casos de uso. |
| CURSO NORMAL DE LOS EVENTOS | |
| Acción del Actor | Respuesta del Sistema |
| | 1. el sistema muestra los servicios facturados de la empresa asociada al auditor. |
| 2. el <actor> una a una los servicios relacionados a una factura que desea auditar. | |
| Pos-Condiciones | Ninguna. |

Tabla 21. Caso de Uso Consultar Factura.

| | |
|---------------------|---------------------------|
| Caso de uso: | Consultar Factura. |
| Autores | Peter Emerson Pinchao. |
| Actores | |



| | |
|--|---|
| Caso de uso: | Consultar Factura. |
| | <ul style="list-style-type: none"> • Usuario • Servicios Web. |
| Pre-Condición: | El <actor> estar revisando el caso de un ingreso o servicion en particular. |
| Descripción: | Este caso de uso inicia cuando el <actor> desea consultar la información de los conceptos de cobro de prestación de servicios de la IPS a un ingreso determinado. |
| CURSO NORMAL DE LOS EVENTOS | |
| Acción del Actor | Respuesta del Sistema |
| 1. el <actor> envía la solicitud de auditar una factura. | |
| | 2. El sistema le muestra los datos de la factura el valor y concepto de cada servicio facturado. |
| Pos-Condiciones | Ninguna. |

F.1.3 MODELO CONCEPTUAL.

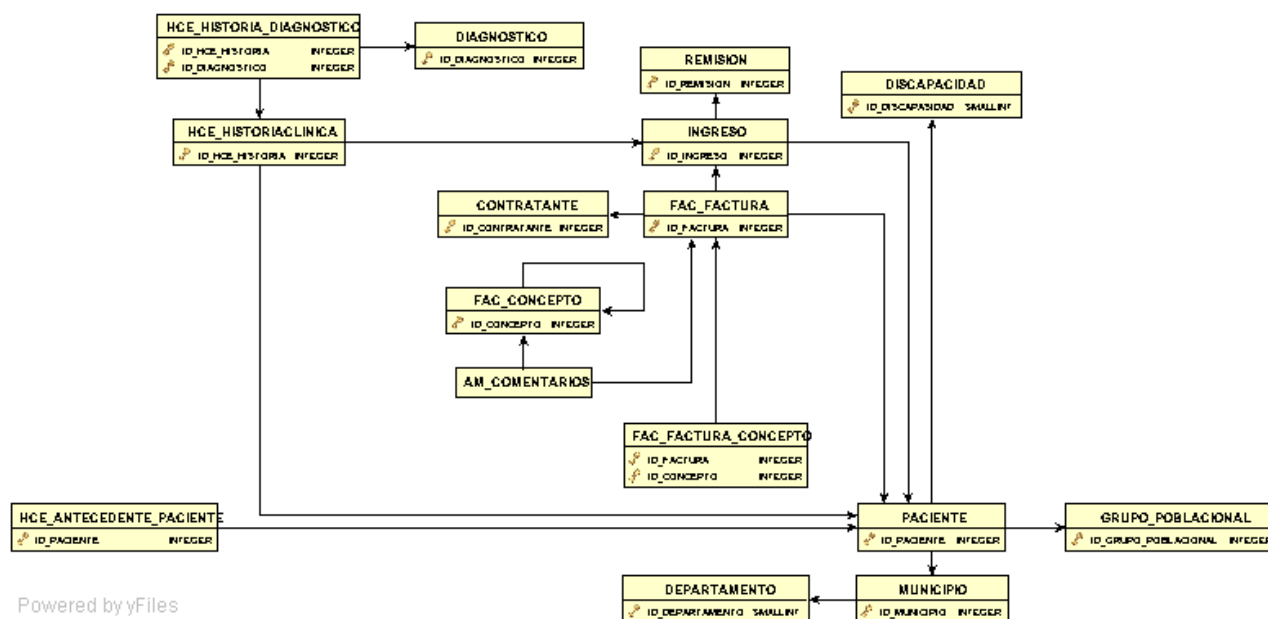


Figura F.3. Modelo conceptual

F.1.4. DIAGRAMA DE SECUENCIAS.

La Figura 61 expone el flujo de la ejecución de los servicios de aplicación donde por defecto se consulta la epicrisis, en caso de requerir más información el servicio de negocio invocará otros servicios separadamente.

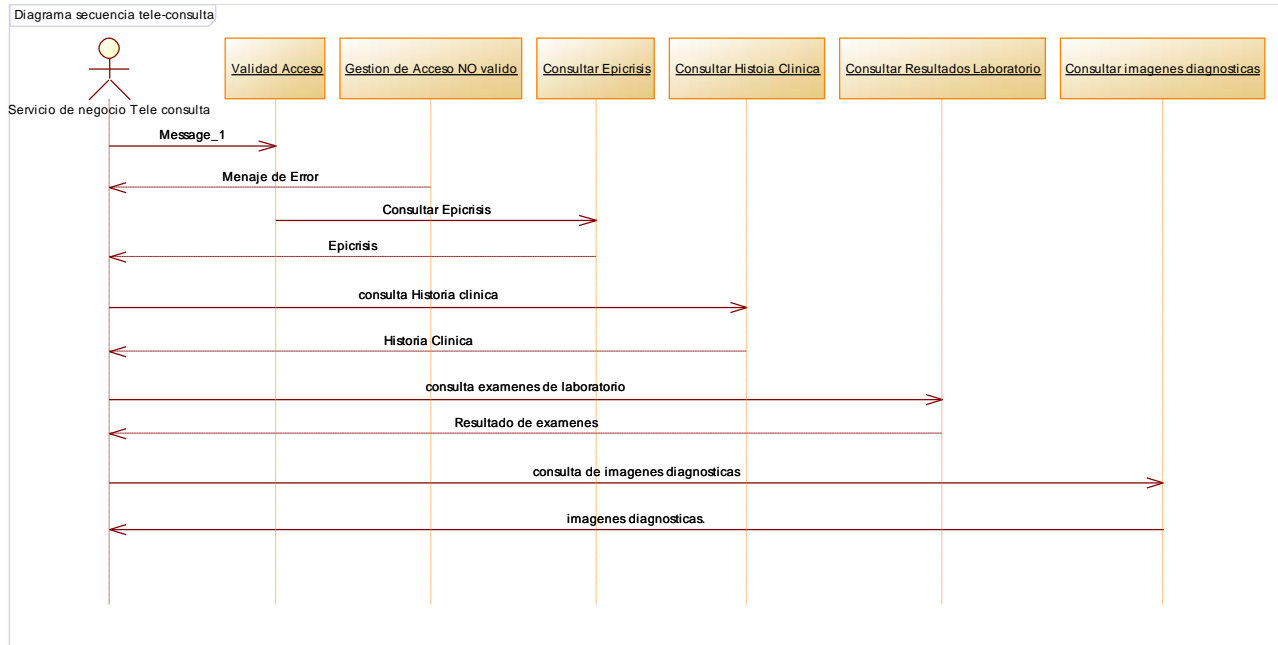


Figura F.4. Diagrama de secuencia servicio tele consulta. flujo comun.

Quando el servicio es invocado con los parámetros necesarios puede retornar otro tipo de respuestas, que incluyan la ejecución y resultados de estas de otros servicios de aplicación. Este caso se expresa en el siguiente diagrama.

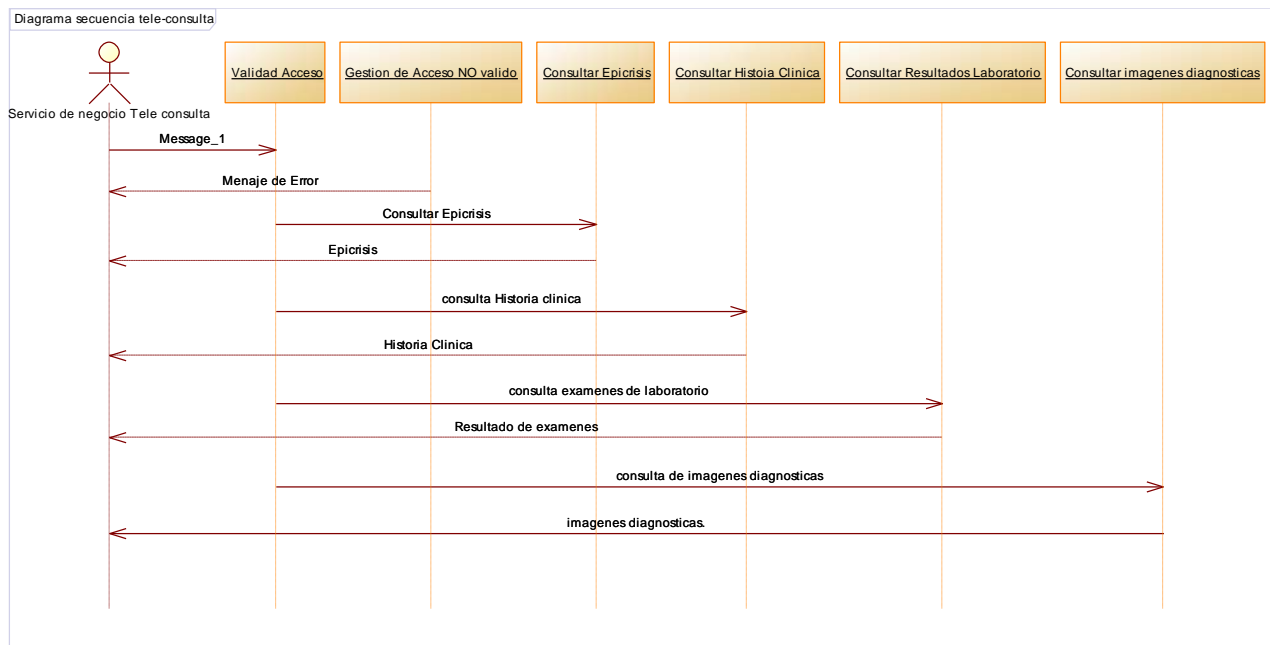


Figura F.5. Diagrama de secuencia servicio de teleconsulta. Parametrizado

De la misma forma se puede representar el diagrama de secuencia de ambas formas para el proceso de auditoría. se presenta a continuación el diagrama cuando se parametriza la invocación del servicio para que habilite la ejecución de otros servicios.

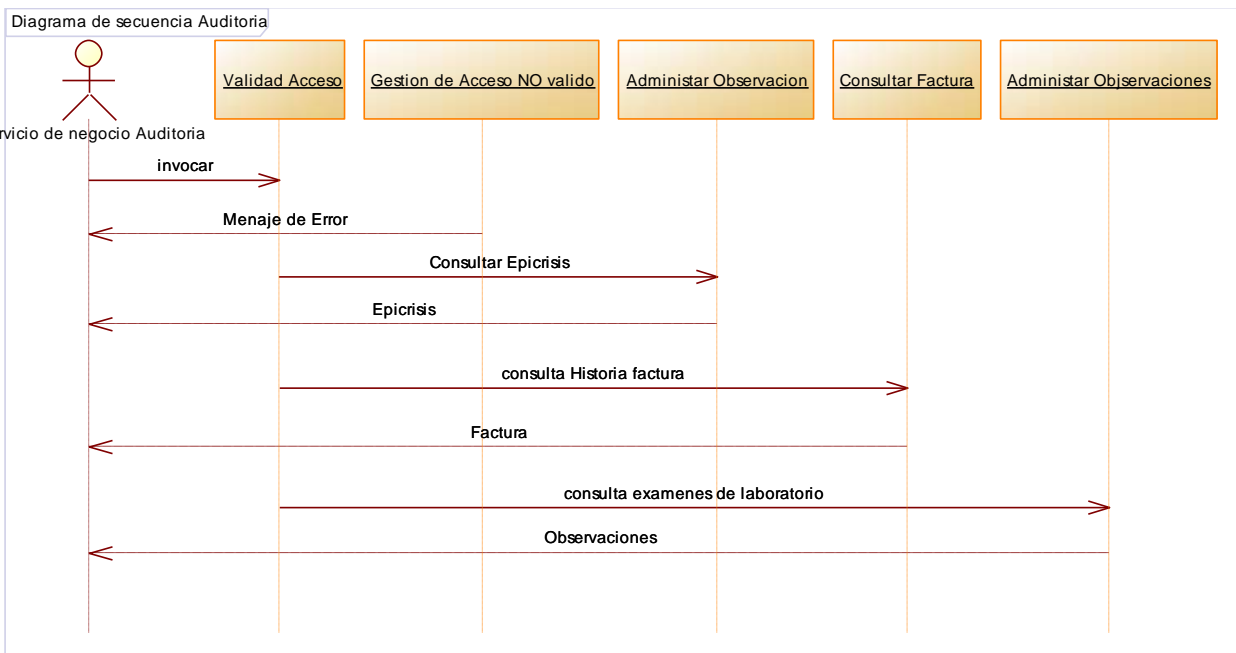


Figura F.6. Diagrama de secuencia de auditoria.

F.2.1. DIAGRAMA DE PAQUETES.

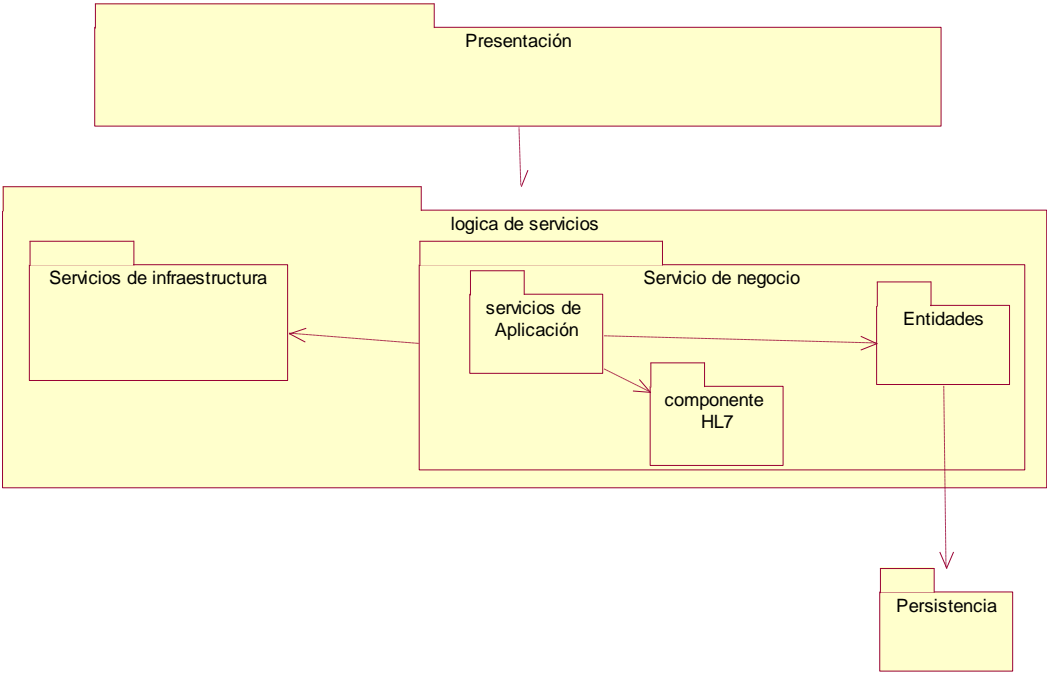


Figura F.7. Diagrama de paquetes

F.2.2. MODELO FÍSICO.

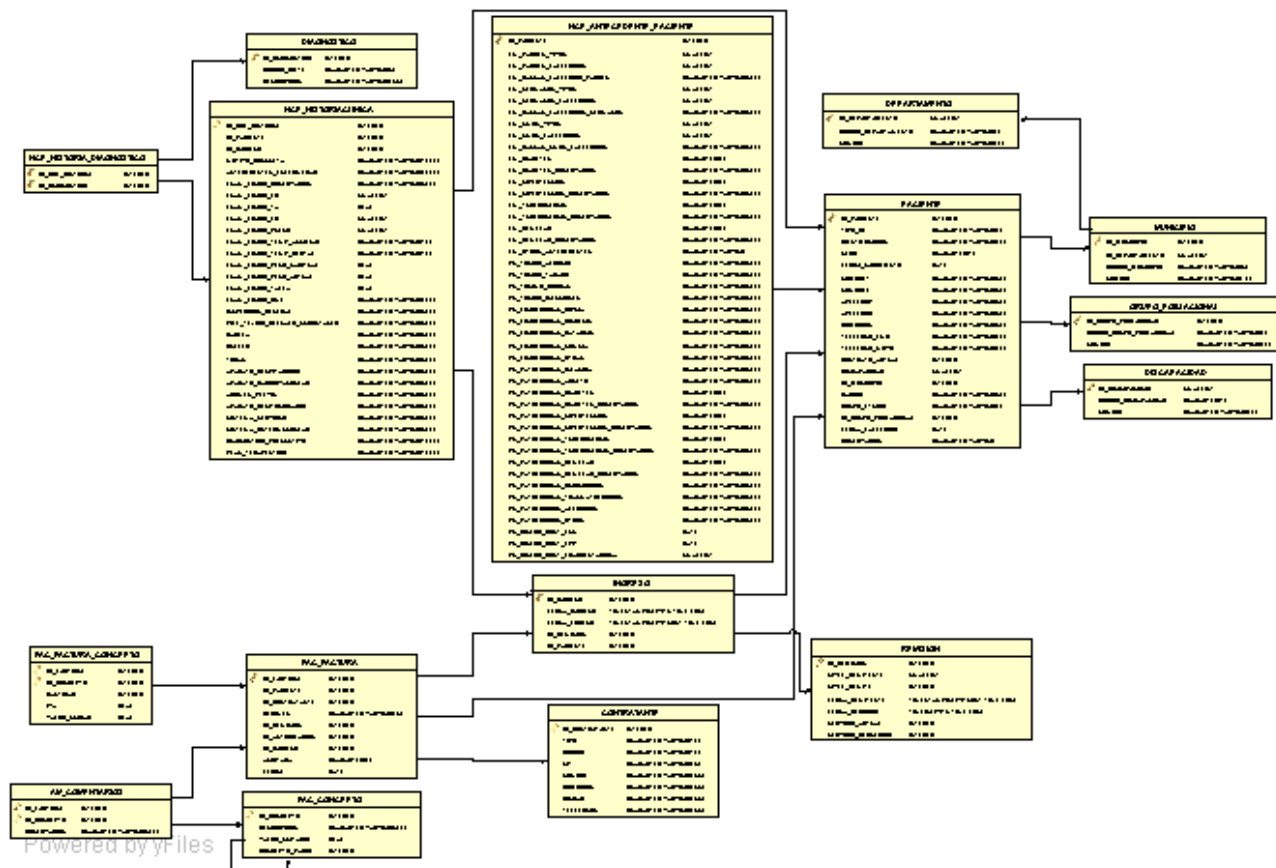


Figura F.8. Modelo físico del prototipo solución.

F.3. IMPLEMENTACIÓN.

Dado que estimo JEE como plataforma tecnológica para desarrollar los aplicativos futuros, se describe en este aparte la configuración de la versión más reciente del IDE Netbeans 7.0 para trabajar con el servidor de ampliaciones instalado Glassfish que soporta el OpenESB.

Se debe ir al menú Tools y el sub Menú Servers, en la parte inferior seleccionar Add Server como se muestra en la siguiente figura.

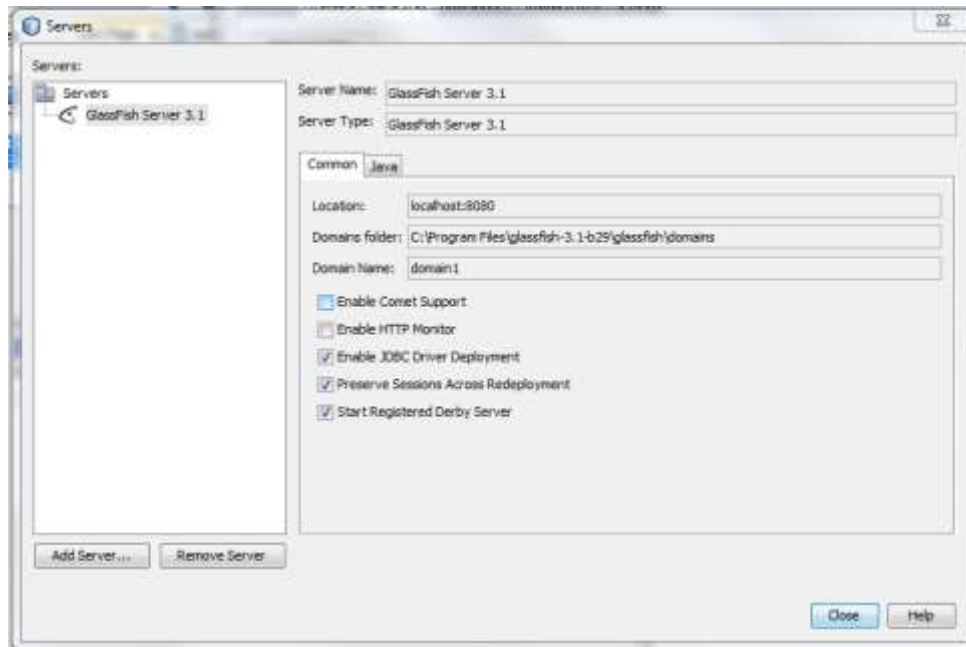


Figura F.9. Administración de Servidores Netbeans.

Luego se selecciona GlassFish 2.x como sigue.

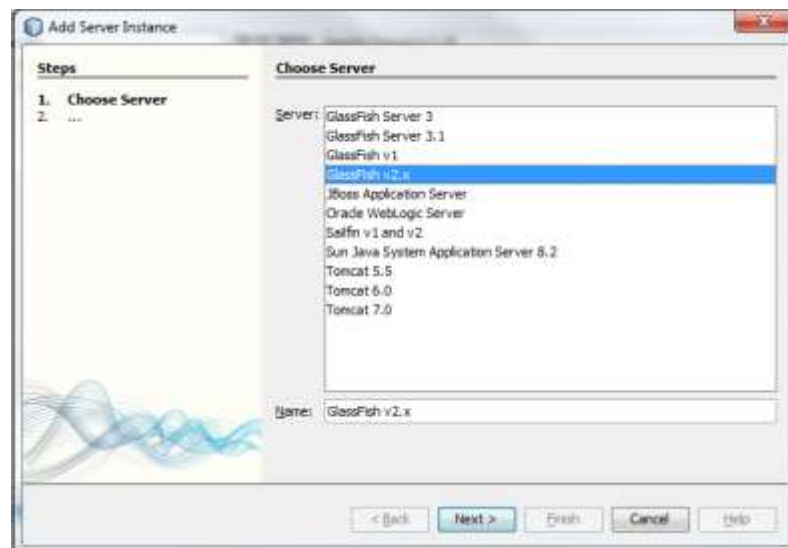


Figura F.10. Selección de GlassFish 2.x.

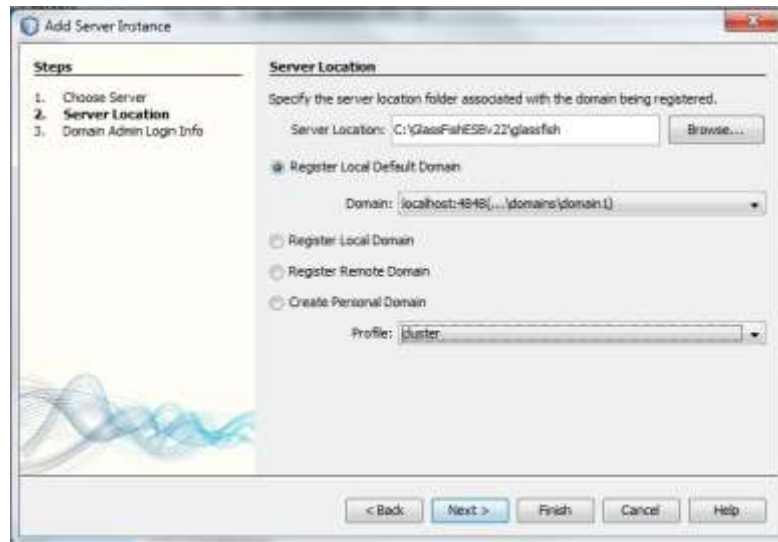


Figura F.11. Configuración de GlassFish en NetBeans 7.0

Como se observa se agrega el servidor con las características de clúster, sin embargo una manera alternativa se describe en el anexo C (C.1.1). Seguidamente se le da la ruta del dominio donde corren las aplicaciones.

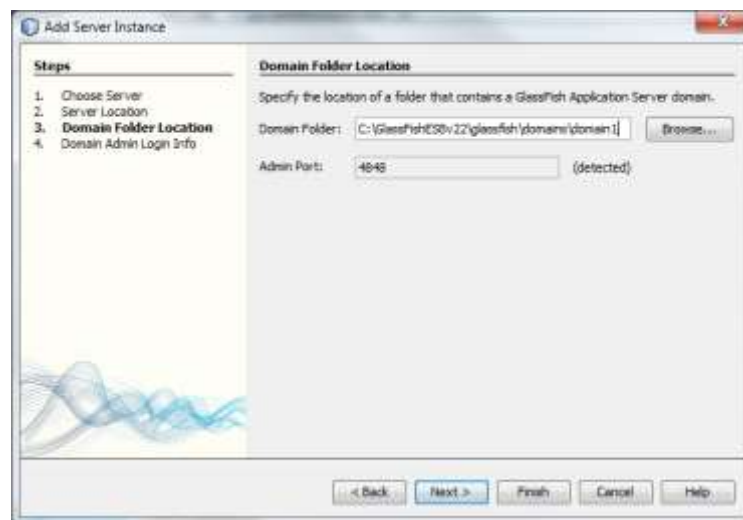


Figura F.12. Configuración del dominio.

A continuación se agregan los datos de autenticación de la consola de administración.



Figura F.13. Sesión de administración.

F.3.1. MÓDULO EJB.

Se debe crear un módulo EJB; para esto lo desde nuestro NetBeans, hacer click en nuevo y luego seleccionar Java EE y dentro de esta opción escoger “EJB Module”, se pone el nombre, del módulo en este caso EJBSoaSiS, seleccionamos como servidor a GlassFish 2.x y finalizar.

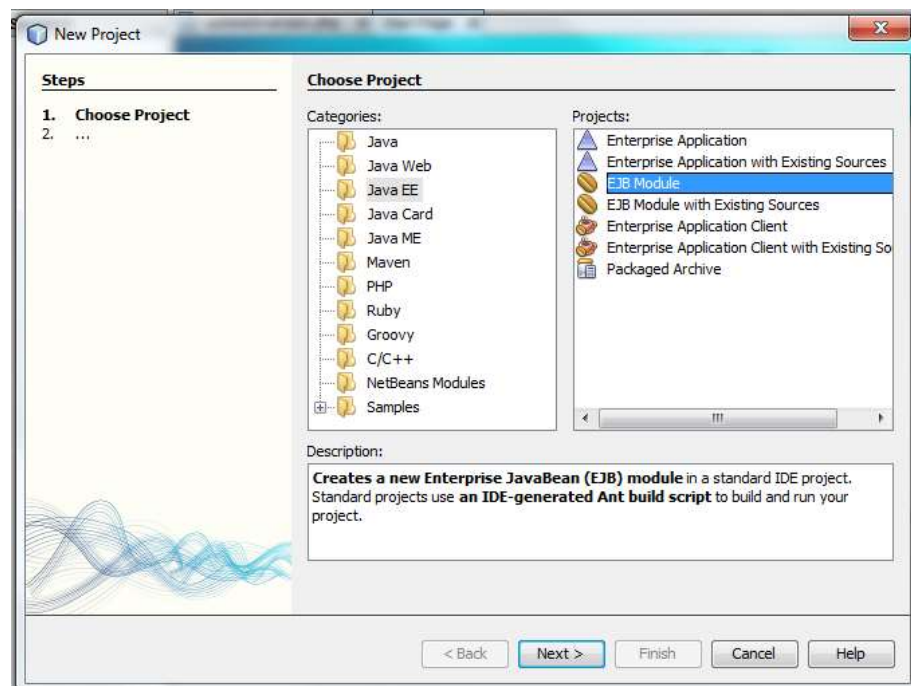


Figura F.14. Crear un Módulo EJB.

Con esto ya se obtiene el módulo EJB listo para crearle toda la lógica necesaria para obtener la información de las tablas la base de datos.

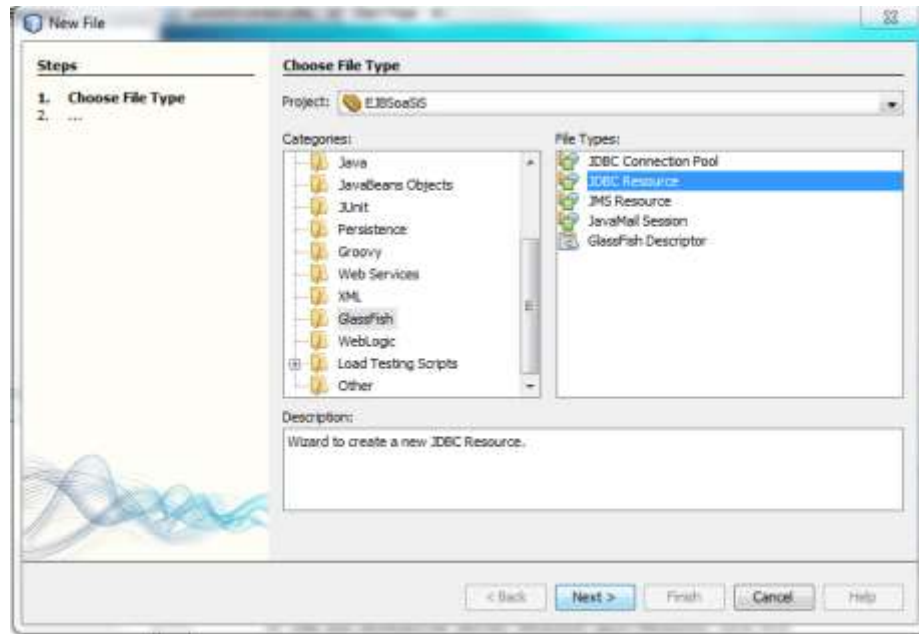


Figura F.16. Selección De JDBC Resource.

Seleccionar el Pool existen, es importante considerar que se debe dar el JNDI correcto que definimos anteriormente como se muestra a continuación.

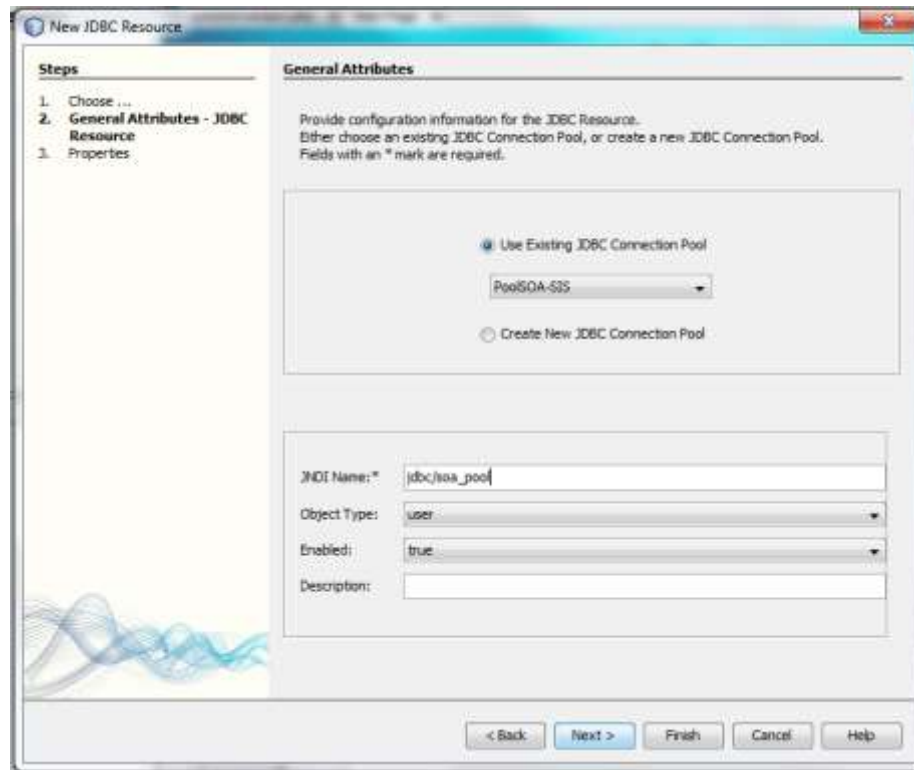


Figura F.17. Selección del Pool de Conexiones Existente.

Ahora se crearán las clases entidad que representaran las tablas de la base de datos. Para esto se debe hacer click derecho en el módulo creado y dirigirse a la opción “Other” y luego en la ventana de opciones, seleccionar “Persistence” y una vez allí seleccionar “Entity Classes From DataBase”.

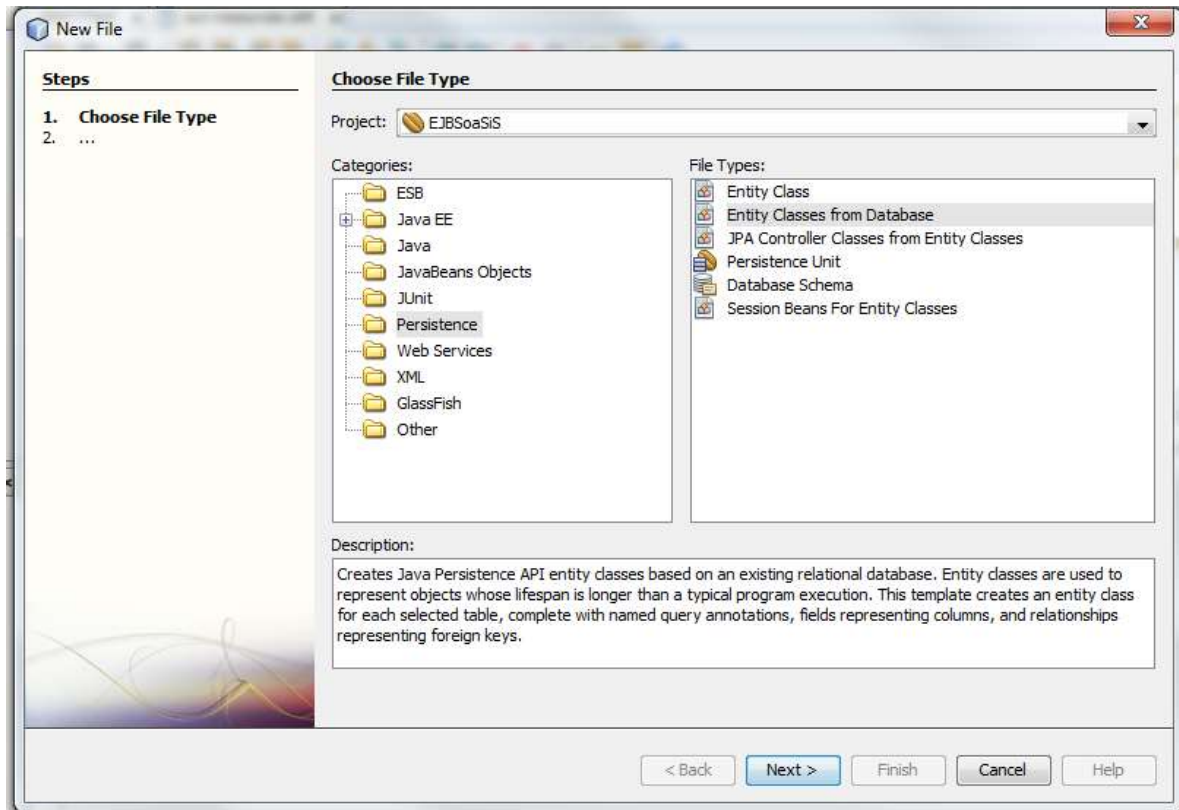


Figura F.18. Seleccionar persistencia mediante JPA.

Se da click en siguiente y aparecerá una ventana en donde se puede ver la opción “Data Source”, se despliega y da click en “New Data source” y luego allí mostrara una pequeña ventana en donde se ingresa el nombre que representa el “Data Source”, que en este caso es “jdbc/soa_pool” y el “Database Connection” que es en donde escogeremos la conexión a la base de datos que se había agregado al inicio.

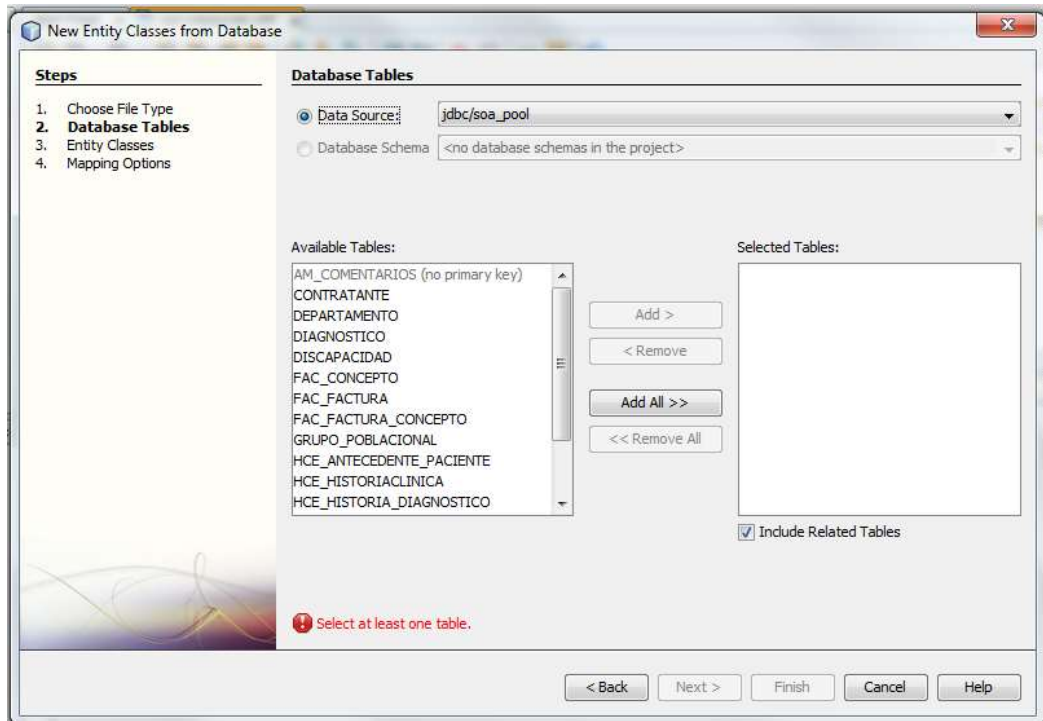


Figura F.19. Selección de la Fuente de datos.

Se seleccionan las tablas a mapear en este caso serán todas dando click al botón “Add All”. Al dar click a continuación se nos muestra una ventana en donde tenemos el nombre de la tabla (lado izquierdo) y el nombre de la clase que representa a esta tabla (lado derecho), el nombre del proyecto, ingresaremos el nombre del paquete en donde se guardara la clase al cual llamaremos **soasis.ejb.entidad**.

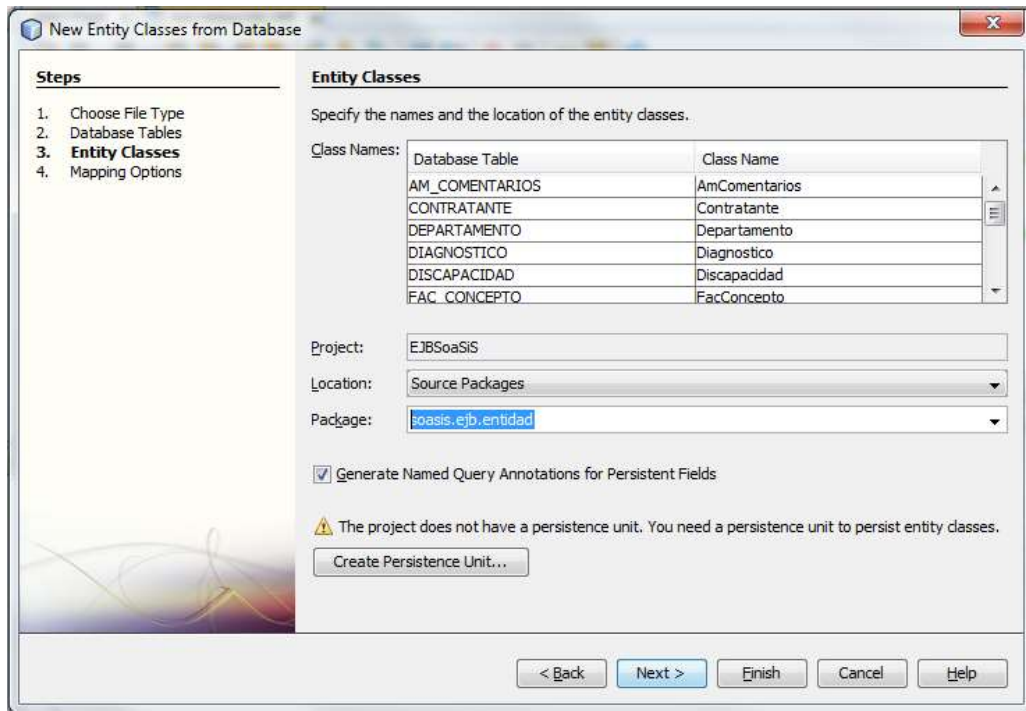


Figura F.20. Selección de las clases y el paquete al que pertenecen.

Al dar siguiente obtenemos el modelo-relacional en clases Java bajo la especificación JPA.

Ahora se creará una unidad de persistencia dando clic derecho en el proyecto “new” luego “Create Persistence Unit”.

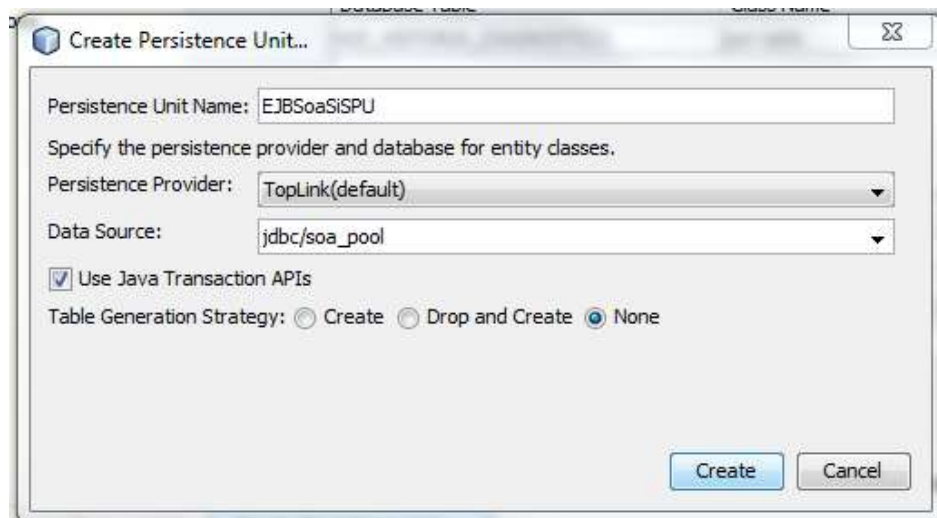


Figura F.21. Creando Unidad de Persistencia.

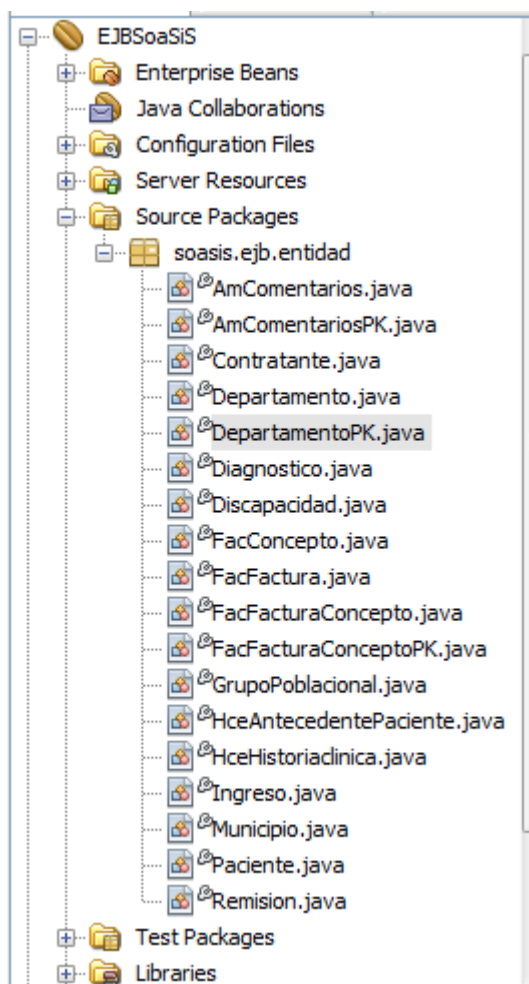


Figura F.22. Modelo Objeto Relacional JAVA.

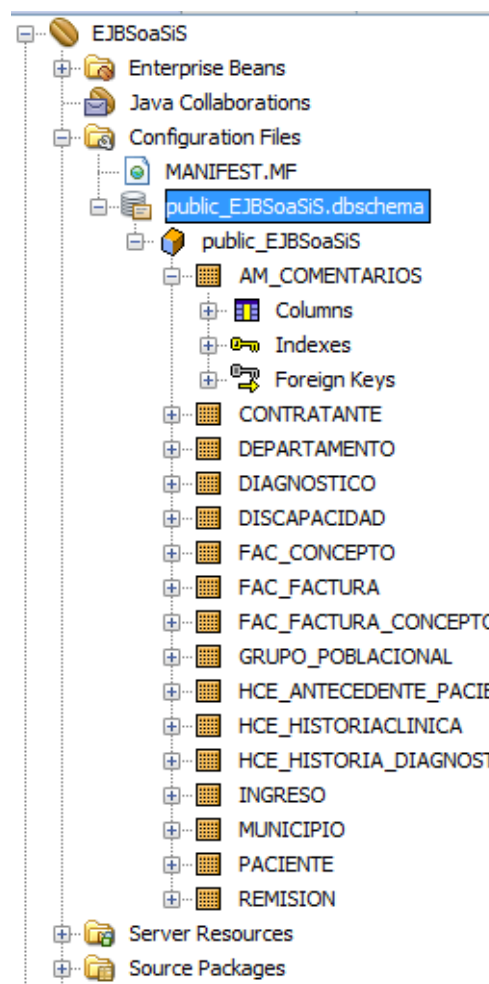


Figura F.23. Modelo de datos Relacional.



Aunque se tiene 16 tablas en la base de datos el modelo representa 18 entidades (18 Clases), dado que las llaves primarias son representadas como la composición de una clase con PK de sufijo en el nombre.

Ahora se crearan los EJB para que suministren la lógica de las entidades de la base de datos.

Se debe darle click derecho al proyecto, New y ubicado en “persistence” y luego seleccionar “Session Beans For Entity Classes”, que como nos podemos dar cuenta, creara las clases en base a la clase que representan las entidades, que en este caso se selecciona todo, hacer click en siguiente y vemos las clases en el panel izquierdo, haciendo click en Add para pasarlo al panel derecho.

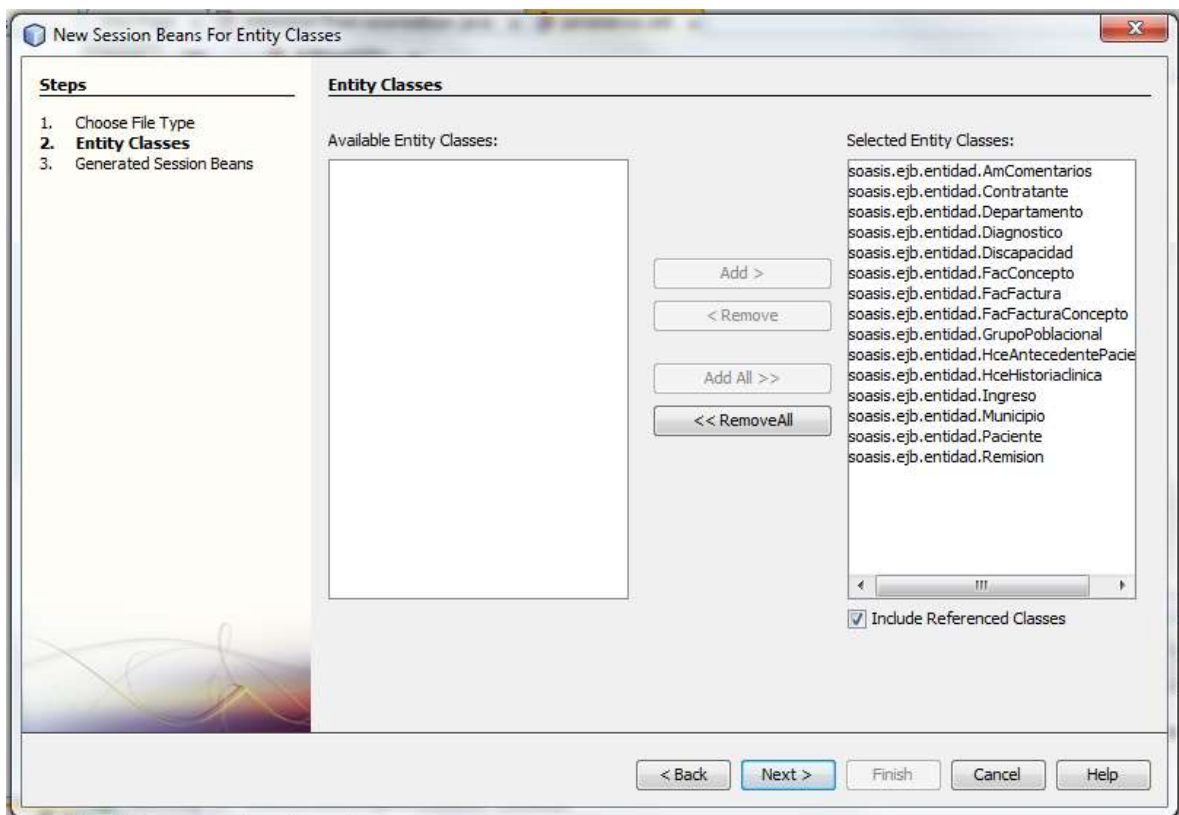


Figura F.24. Seleccionar Entidades.

Luego click en siguiente y aparecerá una ventana con los datos recogidos anteriormente, si gustan el nombre del paquete lo pueden cambiar para de esta forma tener estas clases separadas de las entidades, para este caso se le puede poner **soasis.ejb.control** y luego seleccionamos Remote y Local para el tipo de interface y finalizar.

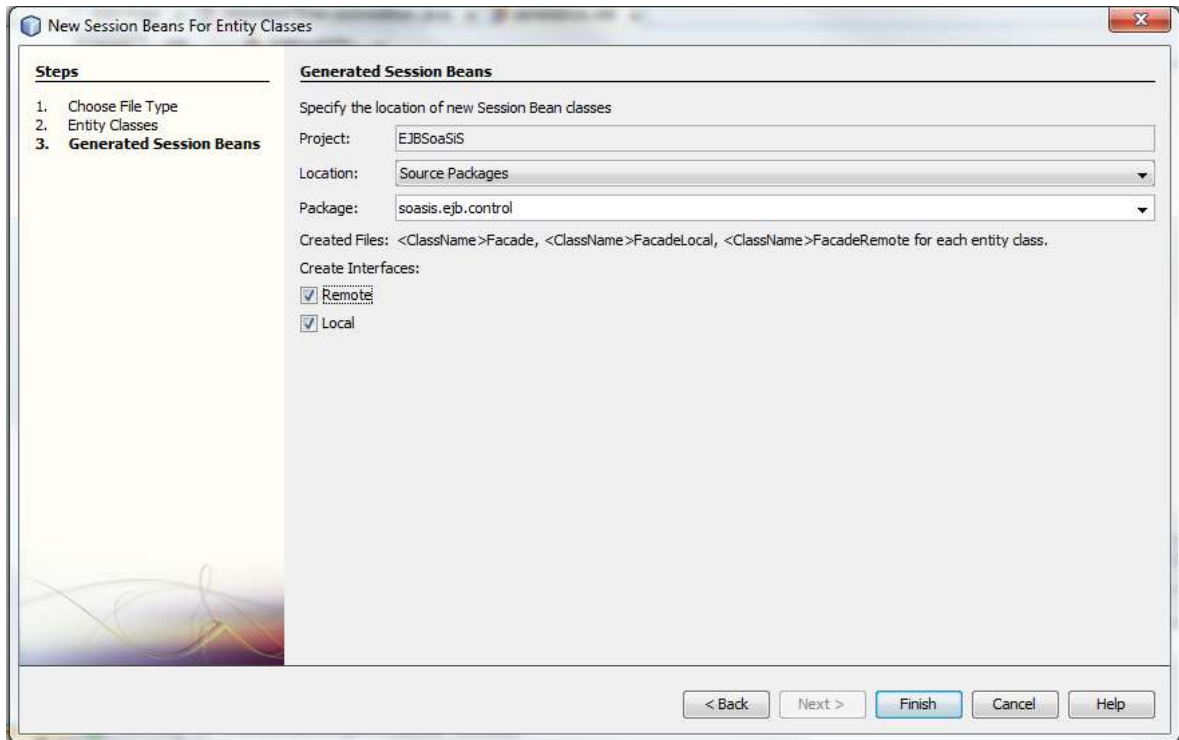


Figura F.25. Crear el Controladores a las entidades como EJBs.

Ahora se pueden ver los EJB como unidades control que soportan las funcionalidades de persistencia.

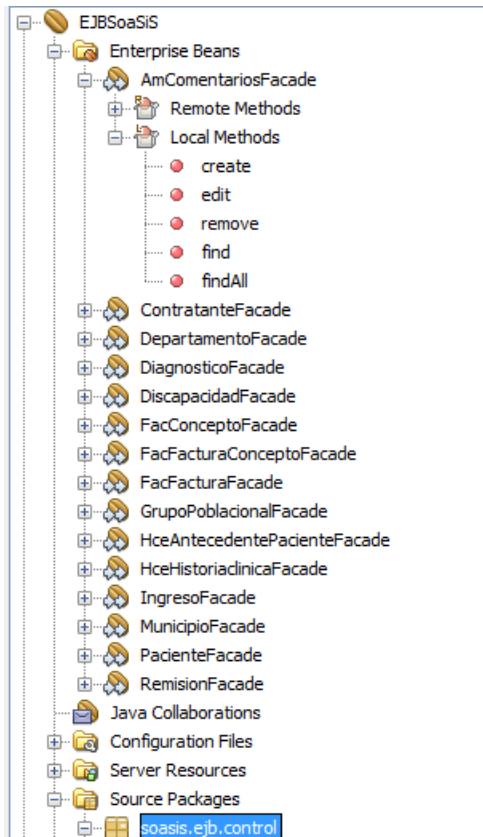


Figura F.26. Entidades como EJB de persistencia.

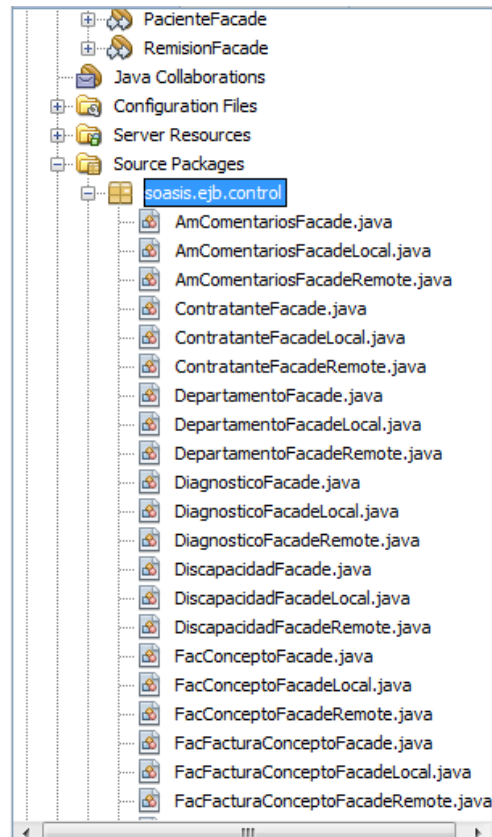


Figura F.27. Fachadas para acceso Local y Remoto.

La Figura F.26 muestra los métodos comunes de persistencia como lo son las operaciones CRUD (Create, Read, Update y Delete) adicionalmente se pueden crear métodos personalizados como consultas personalizadas.

F.3.3. LÓGICA DEL NEGOCIO EN EL MÓDULO EJB SERVICIO DE APLICACIÓN: ADMINISTRAR OBSERVACIONES.

Se ha seleccionado el servicio de Administración de observaciones para describir en este punto como un EJB puede encapsular la lógica de este proceso y como se verá más adelante se convertirá en un servicio Web.

Ahora lo que se hará es crear la clase que implementaran los métodos para toda la lógica de un servicio de aplicación, estas clases serán del tipo SessionBean sin estado dado que no requerimos almacenar información de las operación que el auditor medico hace en el EJB, sino en la Base de Datos.

En orden a lo anterior debe proceder de la siguiente forma:

Darle click derecho al proyecto, clic en New y seleccionar Session Bean....

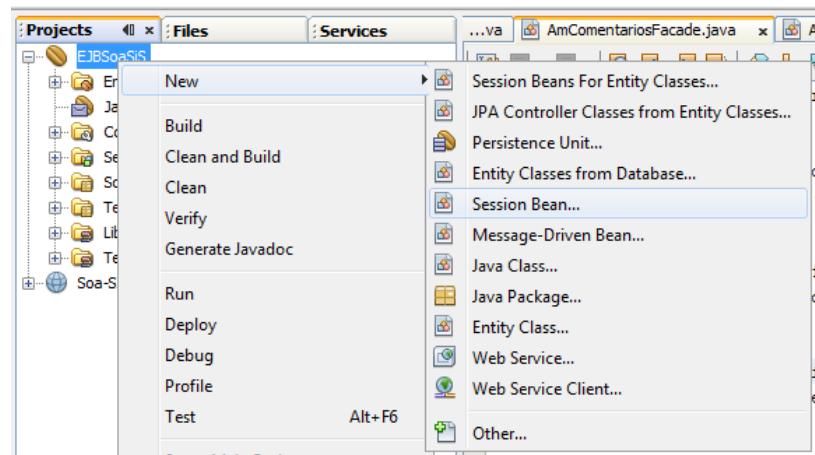


Figura F.28. Crear Un Bean de Sesion.

Luego se escoge el nombre del POJO base del EJB “AdministrarObservaciones” seleccionamos un paquete en este caso **soasis.ejb.auditoriamedica**, seleccionamos el tipo de Bean (sin estado: Stateless) y creamos las Interfaces (Facades) para que sean consumidos los servicios que provee el EJB, en este caso porque pertenecerán a un entorno distribuido de cluster usaremos la interface Remote (para que se puedan emplear a través de RMI los objetos de diferentes instancias del cluster) y la Local (para que emplee en el caso de ejecutarse en el contenedor de EJB local).

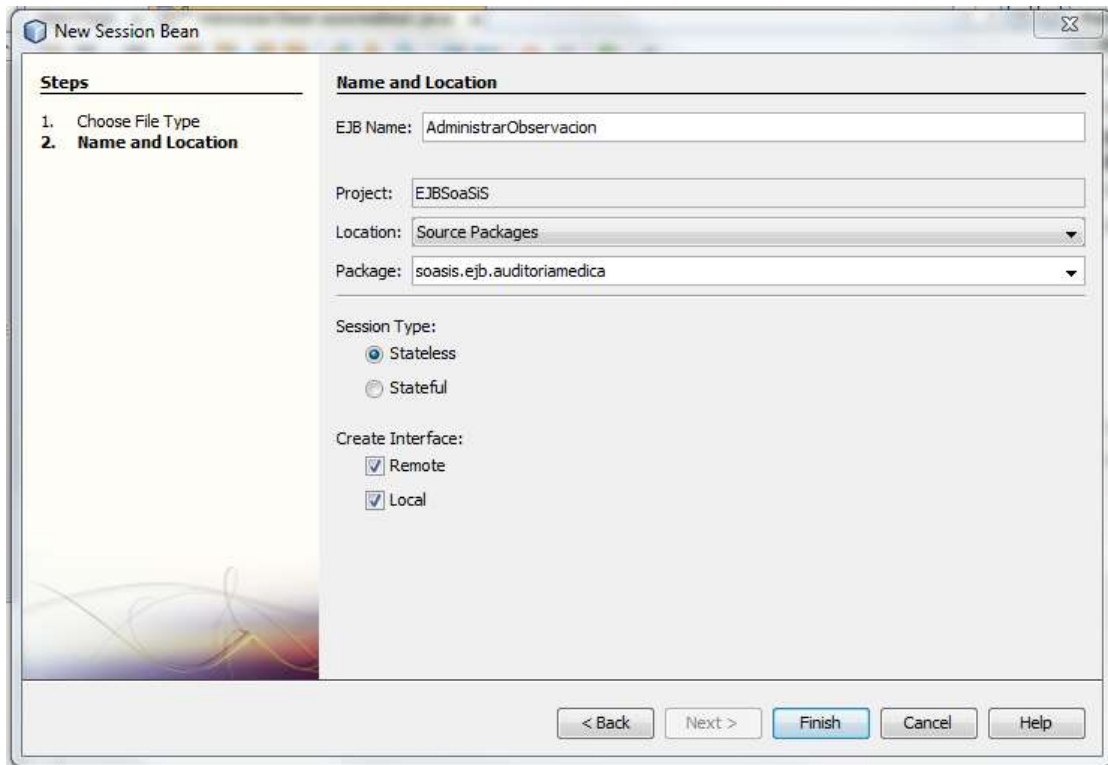


Figura F.29. Crear un Bean de Session Administrar Observaciones.

Con esto tenemos las fachadas (facades) para acceder al EJB y los métodos que define por lo pronto no hay esto se muestra en la Figura F.29.

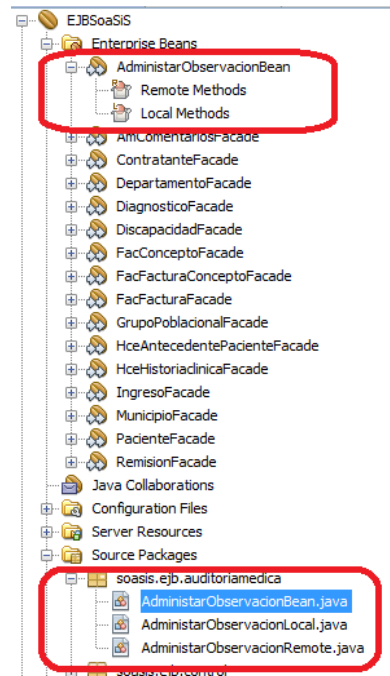


Figura F.30. EJB Administrar Observaciones.

Ahora se puede implementar la lógica del servicio de aplicación, sobre el Bean de sesión y se publican las interfaces en las Fachadas (Facades) haciendo esto se obtiene como se muestra en la Figura F.31.

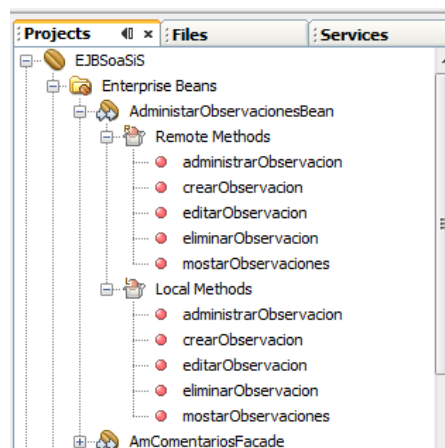


Figura F.31. Bean de Sesión para administrar Observaciones.

Dentro de cada método esta la lógica que invoca los diferentes beans de sección que representan el modelo de datos del sistema.

F.3.4. SERVICIOS WEB MEDIANTE EJB.

Ahora resta publicar las funcionalidades del EJB como un servicio web, para ello se debe proceder como se explica a continuación.



Primero se debe ubicar sobre el módulo EJB creado hacer click derecho seleccionar New y luego Web service.

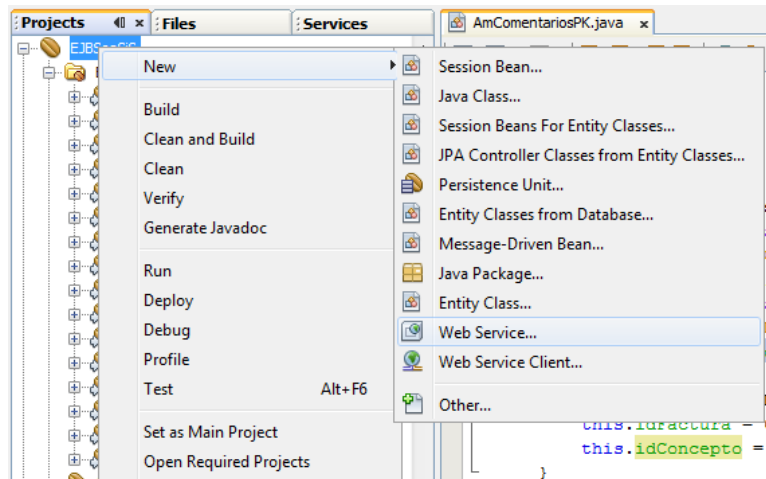


Figura F.32. Agregar nuevo servicio Web.

Posteriormente en el cuadro de dialogo se pone un nombre al servicio web, se selecciona el paquete donde se quiere ubicar y se marca el cuadro “Create Web service from Existing Session Bean”, lo que permitirá en otro cuadro de dialogo seleccionar el Bean de Sesión con el que se creara el servicio web como se muestra en la **¡Error! No se encuentra el origen de la referencia..**

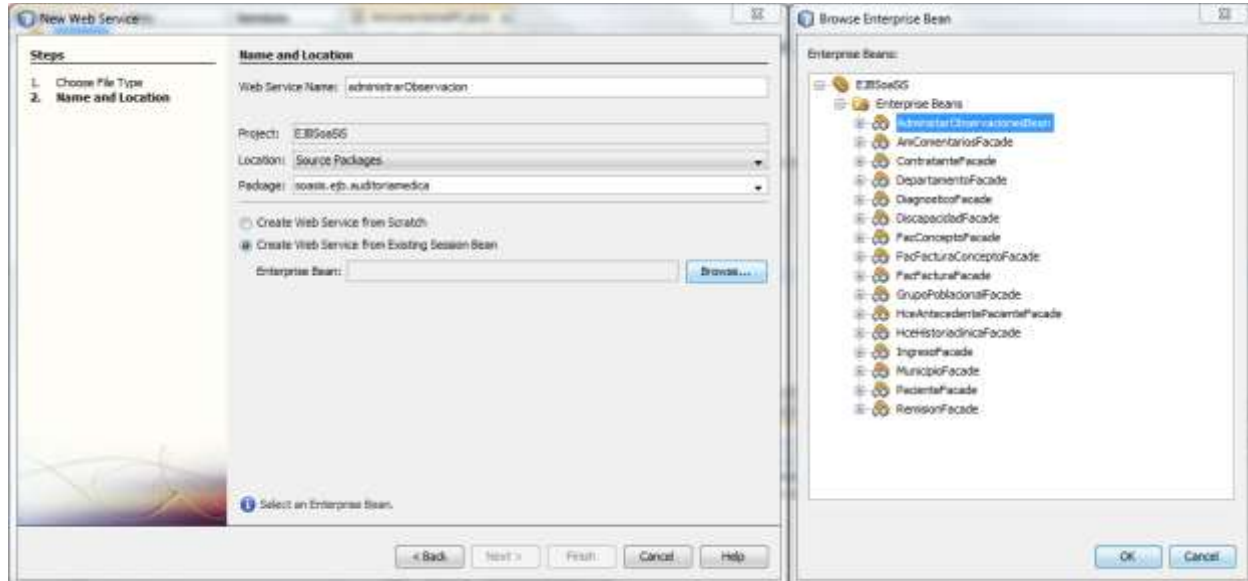


Figura F.33. Crear un Web Service desde un EJB de Sesión.

Lo anterior nos crea un servicio web a partir de un Bean de sesión como se muestra en la

Figura F.34.



```
AmComentariosPK.java x AdministrarObservacionesBean.java x administrarObservacion.java x
Source Design
@Stateless()
public class administrarObservacion {
    @EJB
    private AdministrarObservacionesLocal ejbRef;

    public administrarObservacion() {
    }
    // Add business logic below. (Right-click in editor and choose
    // "Web Service > Add Operation"

    @WebMethod(operationName = "administrarObservacion")
    public String administrarObservacion(@WebParam(name = "operacion")
    String operacion, @WebParam(name = "idFactura")
    int idFactura, @WebParam(name = "idConcepto")
    int idConcepto, @WebParam(name = "observacion")
    String observacion) {
        return ejbRef.administrarObservacion(operacion, idFactura, idConcepto, observacion);
    }

    @WebMethod(operationName = "crearObservacion")
    public String crearObservacion(@WebParam(name = "idFactura")
    int idFactura, @WebParam(name = "idConcepto")
    int idConcepto, @WebParam(name = "observacion")
    String observacion) {
        return ejbRef.crearObservacion(idFactura, idConcepto, observacion);
    }
}
```

Figura F.34. Web Service desde un EJB de Session.



G PRUEBAS DE RENDIMIENTO Y ESCALABILIDAD.

Características de los equipos.

Tabla 22. Pruebas de Rendimiento.

| equipo | IP | SO | RAM | HD | CPU |
|-------------|---------------|--------------------|-------|-------|---------------|
| OpenESB 1 | 172.16.25.160 | Windows XP Pro | 2GB | 80GB | 1,6 GHz |
| OpenESB 2 | 172.16.25.4 | Windows XP Pro | 2GB | 80GB | 2,6 GHz |
| OpenESB 3 | 172.16.25.3 | Windows XP Pro | 1GB | 30GB | 2,6 GHz |
| Base Datos | 172.16.25.5 | CentOS 5.5 | 2GB | 30GB | 1,7 GHz x 2n |
| Carga 1 | 172.16.25.12 | Windows 7 Ultimate | 4GB | 500GB | 1,65 GHz x 2n |
| Carga 2 | 172.16.25.208 | Windows XP Pro | 512MB | 30GB | 1,6 GHz |
| Balanceador | 172.16.25.32 | CentOS 5.5 | 512MB | 40GB | 1,6 GHz |

G.1. CONFIGURACIÓN DE CLÚSTER OPENESB Y PRUEBAS DE ESCALABILIDAD.

Para realizar las pruebas de escalabilidad se requiere la implementación de una arquitectura distribuida de procesamiento, se describe como se procede a configurar una clúster de OpenESB, en un cluster cada nodo debe tener la configuración que los demás.

La siguiente imagen describe la arquitectura de cluster:

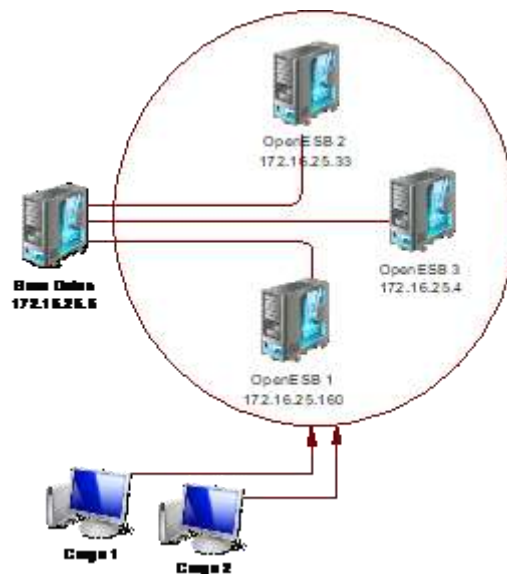


Figura G.1. Diagrama de Arquitectura Clúster.

G.1.1. CONFIGURACIÓN DE GLASSFISH SOPORTE DE CLÚSTER.



Se describe como se le configuran dos instancias en el servidor OpenESB 1 que es el controlador del cluster seguimos estos pasos:

Descargamos e instalamos el GlassFish ESB v2.2 con las mismas características y los mismos servicios en los tres equipos destinados para tal fin. Y se los pone en marcha

Una vez puestos en marcha los servidores de aplicaciones GlassFish se inicia sesión en la consola de administración en <http://localhost:4848> o <http://172.16.25.160:4848>, utilizando el nombre de usuario por defecto: admin y la contraseña: adminadmin.

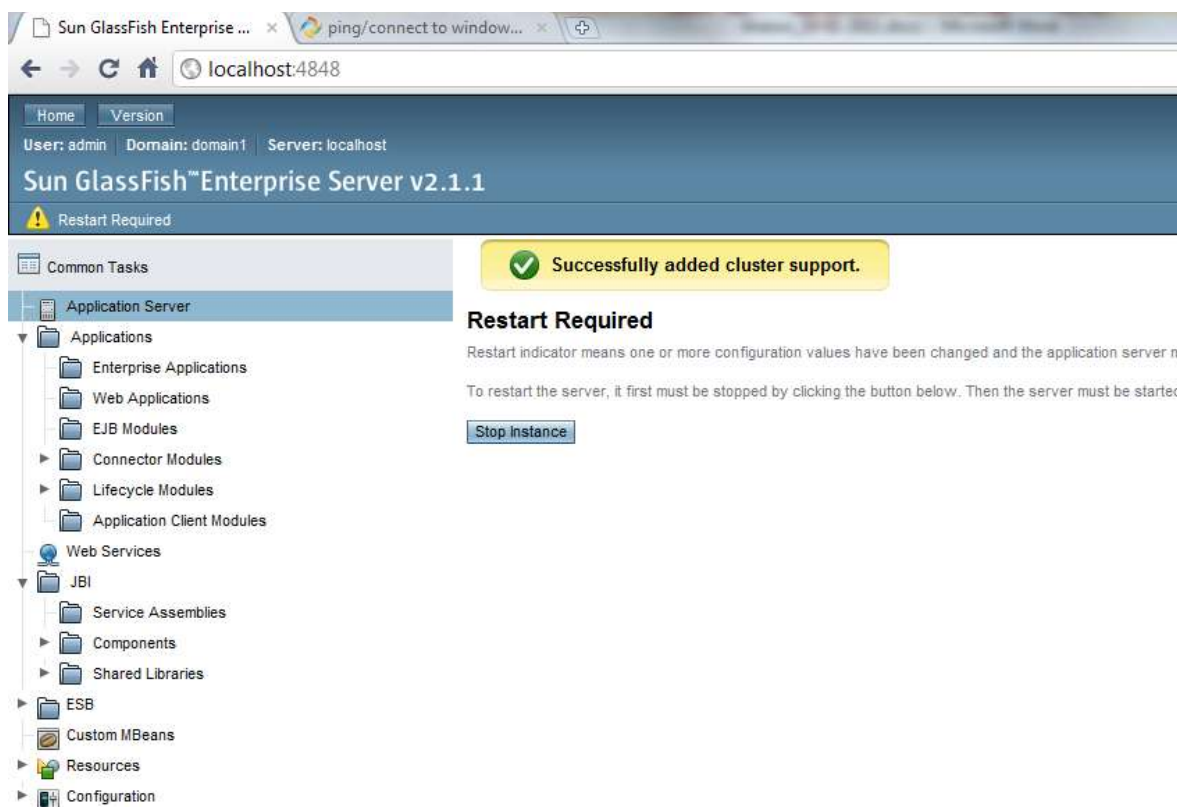
Desde la consola de administración, haga clic en add Cluster Support. Revise los Añadir clúster página de soporte y haga clic en Aceptar.



Damos click en Ok.



Detenemos la instancia de GlassFish, y luego la volvemos a iniciar de manera que quede activo el soporte para cluster.



Entramos a una interfaz de comandos de línea (Comman Line Interface, CLI) y continuación se entra en el directorio bin que se encuentra en el home de GlassFish, en este caso sería:

C:\GlassFishESBv22\glassfish\bin



Escribimos el comando que permite entrar a la consola como usuario administrador:

```
... bin>asadmin.
```

Crea un agente que llamaremos nodoESB2 con el comando:

```
asadmin> create-node-agent --host 172.16.25.160 --port 4848 nodoESB2
```

Una guía paso a paso a partir de aquí se encuentra en video en el siguiente enlace:
<http://dlc.sun.com.edgesuite.net/javaee5/screencasts/createCluster/>

G.1.2. ESCENARIOS DE EVALUACIÓN DE LA ESCALABILIDAD.

Siendo la evaluación comparativa la esencia de una evaluación de escalabilidad se comparan dos sistemas informáticos. El caso más ideal sería comparar los servicios de negocio de la SOA solución con una aplicación existente en el HSLV bajo JEE que realice los mismos procesos de negocio. Desafortunadamente en el HSLV no existen tales sistemas que permitan hacer esta comparación, de esta forma se empleara una aplicación de la empresa ASMET Salud EPS de auditorías médicas, llamada AMedicas (Auditorias Medicas) con la salvedad que la consulta de epicrisis se obtendrá de las bases de datos de Dinámica Gerencial (el mismo del HSLV) para que se logre el objetivo de evaluar el servicio en un entorno similar al del HSLV. La consulta a bases de datos de Dinámica representa un cambio en las fuentes de AMedicas, (permitiendo estos cambios establecer una base comparativa de la flexibilidad como se verá más adelante). De esta forma se puede establecer una comparación de un ambiente SOA y otro No SOA.

Adicional a esta comparación se establecen indicadores de rendimiento claves para establecer el benchmarking, como punto de referencia para hacer una evaluación comparativa y de esta forma tener dos comparaciones una entre ambientes SOA, no SOA y establecer acuerdos de nivel esperados.

Para la evaluación se dispone de los recursos que nos permiten la implementación de cluster del ESB. La evaluación de la arquitectura fue desarrollada con los recursos tecnológicos y logísticos de ASMET Salud EPS-S. El detalle de los escenarios de evaluación son descritos en el Anexo G.

Pruebas de carga.

La primera evaluación sobre los servicios se hará con un host donde se despliegan los servicios de ESB, y un elemento de carga (otros hosts) con los parámetros de carga para tal fin. Antes presentar los valores de carga se describen los parámetros de configuración de los algoritmos de carga.

- **Hilos:** Cada hilo que se configura es un proceso de ejecución del proceso de negocio.
- **Periodo ente Hilos:** Define cada cuantos segundos se disparará cada Hilo (conurrencia).
- **Número de veces de ejecución:** cuantas veces se ejecutaran determinado número de Hilos (recurrencia).



En la Tabla 32 se describen los parámetros de carga y la configuración de clúster y de carga para la prueba:

Tabla 23. Pruebas de carga.

| ID Prueba | Nodos del Cluster | Host de carga | Numero de Hilos por grupo | periodo entre grupo de Hilos (seg) | Número ejecuciones |
|-----------|-------------------|---------------|---------------------------|------------------------------------|--------------------|
| P.2.1 | 1 | 1 | 120 | 30 | 3 |
| P.2.2 | | 1 | 180 | 20 | 6 |
| P.2.3 | | 1 | 240 | 10 | 9 |
| P.2.4 | 2 | 2 | 240 | 20 | 3 |
| P.2.5 | | 2 | 380 | 10 | 6 |
| P.2.6 | | 2 | 500 | 5 | 9 |
| P.2.7 | 3 | 2 | 600 | 20 | 3 |
| P.2.8 | | 2 | 900 | 10 | 6 |
| P.2.9 | | 2 | 1000 | 5 | 9 |

Con esas pruebas podemos establecer la evaluación de las pruebas de carga. Sin embargo como se desea establecer una comparación entre ambientes SOA y no SOA se aplicara el mismo modelo de prueba a la solución AMedica de la Empresa ASMET Salud (considerando P.3.x las pruebas de punto inicial y P.4.x las pruebas de carga).

Resultados de estas pruebas son:

Tabla 24. Resultados de pruebas de carga

| Nodos Cluster | ID Prueba | Latencia (ms) | Throughput (TPS) | Disco Duro % | | | Memoria RAM (%) | | |
|---------------|-----------|---------------|------------------|--------------|--------|-------|-----------------|-------|-------|
| 1 | P.2.1 | 1933 | 273 | 36,3 | | | 86,6 % | | |
| | P.2.2 | 2223 | 452 | 36,7 | | | 88,3 % | | |
| | P.2.3 | 2536 | 467 | 39,9 | | | 91,1% | | |
| 2 | P.2.4 | 1702 | 587 | 46,8% | 38,3 % | 91,2% | 82,2% | | |
| | P.2.5 | 1493 | 522 | 52,2% | 38,7% | 92,8% | 74,6% | | |
| | P.2.6 | 1587 | 656 | 46,2% | 39,9% | 84,2% | 65,9% | | |
| 3 | P.2.7 | 1202 | 684 | 66,2% | 45,6% | 35,2% | 92,2% | 72,7% | 82,5% |
| | P.2.8 | 1194 | 721 | 62,1% | 58,2% | 41,6% | 96,5% | 81,3% | 61,2% |
| | P.2.9 | 1025 | 735 | 62,1% | 56,2% | 44,3% | 93,3% | 63,8% | 62,1% |

Pruebas de stress.

Las siguientes progresiones matemáticas se emplearán para determinar la prueba de estrés sobre la arquitectura a fin de determinar los límites del sistema manteniendo constante el periodo entre hilos con un valor de 5 segundos.

- Numero de hilos por grupo: $60 + \sum_{i=1}^n 60 * i$
- Número de ejecuciones: $\sum_i^n 3i$



Tabla 25. Pruebas de Stress.

| ID Prueba | Nodos del Cluster | Host de carga | Numero de Hilos por grupo | periodo entre grupo de Hilos (seg) | Número de veces ejecución |
|-----------|-------------------|---------------|----------------------------|------------------------------------|---------------------------|
| P.5.1 | 1 | 1 | 120 | 5 | 3 |
| P.5.2 | | 1 | 180 | 5 | 6 |
| P.5.3 | | 1 | 240 | 5 | 9 |
| .. | | ... | ... | 5 | ... |
| P.5.i | | ... | $60 + \sum_{i=1}^n 60 * i$ | 5 | $\sum_i^n 3i$ |
| ... | | ... | ... | 5 | ... |
| P.5.n | | ... | $60 + \sum_{i=1}^n 60 * n$ | 5 | $\sum_i^n 3n$ |

La tabla 9 describe la prueba P.5.n donde más del 20% de la ejecución del proceso produce error de denegación de servicio. De esta misma forma se procederá con las pruebas cuando en el clúster existen 2 nodos (para lo cual el ID de la prueba será P.6.x y pruebas P.7x para 3 nodos).

El trabajo de diseñar y ejecutar pruebas de flexibilidad es una ardua labor que requiere de personal experto para determinar en el servidor de aplicaciones (Glassfish) diversas configuraciones de los nodos del clúster (número de sesiones concurrentes, número de conexiones del pool de conexiones, memoria de los objetos en el servidor entre otros) para poder terminar las causas de la denegación de servicio. También se requiere la colaboración de Administradores de Bases de Datos (data Base Administrador, DBA) para gestionar los recursos técnicos de conexiones y disponibilidad de los sistemas de almacenamiento.

Los resultados de estas pruebas son:

Para la el clúster con un nodo.

Tabla 26 resultados de prubas de estres para cluster con un nodo.

| ID Prueba | Nodos del Cluster | Host de carga | Numero de Hilos por grupo | periodo entre grupo de Hilos (seg) | Número de veces ejecución | Denegación de servicio | Número total de hilos | Tiempo de a prueba (seg) | Nivel de la prueba |
|-----------|-------------------|---------------|---------------------------|------------------------------------|---------------------------|------------------------|-----------------------|--------------------------|--------------------|
| P.5.1 | 1 | 1 | 120 | 5 | 3 | 0.0 | 720 | 30 | 24,00 |
| P.5.2 | | | 180 | | 6 | 0. | 1620 | 45 | 36,00 |
| P.5.3 | | | 240 | | 9 | 0. | 2880 | 60 | 48,00 |
| P.5.4 | | | 300 | | 12 | 0.0 | 4500 | 75 | 60,00 |
| P.5.5 | | | 360 | | 15 | 0.0 | 6480 | 90 | 72,00 |
| P.5.6 | | 2 | 420 | | 18 | 0.0 | 8820 | 105 | 84,00 |
| P.5.7 | | | 480 | | 21 | 0.02 | 23040 | 120 | 192,00 |
| P.5.8 | | | 540 | | 24 | 0,3 | 29160 | 135 | 216,00 |
| P.5.9 | | | 600 | | 27 | 2,1 | 36000 | 150 | 240,00 |
| P.5.10 | | | 660 | | 30 | 2,4 | 43560 | 165 | 264,00 |

Propuesta para la Interoperabilidad de Procesos de Negocio en Sistemas de Información Hospitalarios Bajo una Arquitectura Orientada a Servicios



| | | | | | | | | | |
|--------|------|------|------|-----|-------|---------|---------|----------|----------|
| P.5.11 | 3 | 720 | | 33 | 3,1 | 51840 | 180 | 288,00 | |
| P.5.12 | | 780 | | 36 | 3,3 | 60840 | 195 | 312,00 | |
| P.5.13 | | 840 | | 39 | 12,1 | 105840 | 210 | 504,00 | |
| P.5.14 | | 900 | | 42 | 13,7 | 121500 | 225 | 540,00 | |
| P.5.15 | | 960 | | 45 | 16,6 | 138240 | 240 | 576,00 | |
| P.5.16 | | 1020 | | 48 | 21,30 | | | | |
| P.6.20 | 2 | 1260 | 5 | 60 | 3,15 | 156060 | 255 | 612,00 | |
| P.6.21 | | 1320 | 5 | 63 | 2,45 | 238140 | 315 | 756,00 | |
| P.6.22 | | 1380 | 5 | 66 | 3,2 | 285660 | 345 | 828,00 | |
| P.6.23 | | 1440 | 5 | 69 | 3,45% | 414720 | 360 | 1.152,00 | |
| P.6.24 | | 1500 | 5 | 72 | 4,16 | 450000 | 375 | 1.200,00 | |
| P.6.25 | | 1560 | 5 | 75 | 6,12 | 486720 | 390 | 1.248,00 | |
| P.6.26 | | 1620 | 5 | 78 | 6,12 | 524880 | 405 | 1.296,00 | |
| P.6.27 | | 1680 | 5 | 81 | 6,45 | 564480 | 420 | 1.344,00 | |
| P.6.28 | | 1740 | 5 | 84 | 8,65 | 605520 | 435 | 1.392,00 | |
| P.6.29 | | 1800 | 5 | 87 | 9,12 | 648000 | 450 | 1.440,00 | |
| P.6.30 | | 1860 | 5 | 90 | 7,34 | 691920 | 465 | 1.488,00 | |
| P.6.31 | | 1920 | 5 | 93 | 6,23 | 737280 | 480 | 1.536,00 | |
| P.6.32 | | 1980 | 5 | 96 | 5,13 | 784080 | 495 | 1.584,00 | |
| P.6.33 | | 2040 | 5 | 99 | 16,20 | 832320 | 510 | 1.632,00 | |
| P.6.34 | | 2100 | 5 | 102 | 14,20 | 882000 | 525 | 1.680,00 | |
| P.6.35 | | 2160 | 5 | 105 | 23,20 | 933120 | 540 | 1.728,00 | |
| P.7.45 | | 3 | 2760 | 5 | 135 | 4,20 | 1523520 | 690 | 2.208,00 |
| P.7.46 | | | 2820 | 5 | 138 | 6,20 | 1590480 | 705 | 2.256,00 |
| P.7.47 | 2880 | | 5 | 141 | 6,20 | 1658880 | 720 | 2.304,00 | |
| P.7.48 | 2940 | | 5 | 144 | 6,20 | 1728720 | 735 | 2.352,00 | |
| P.7.49 | 3000 | | 5 | 147 | 7,70 | 1800000 | 750 | 2.400,00 | |
| P.7.50 | 3060 | | 5 | 150 | 14,20 | 1872720 | 765 | 2.448,00 | |
| P.7.51 | 3120 | | 5 | 153 | 10,60 | 1946880 | 780 | 2.496,00 | |
| P.7.52 | 3180 | | 5 | 156 | 12,80 | 2022480 | 795 | 2.544,00 | |
| P.7.53 | 3240 | | 5 | 159 | 16,65 | 2099520 | 810 | 2.592,00 | |
| P.7.54 | 3300 | | 5 | 162 | 17,80 | 2178000 | 825 | 2.640,00 | |
| P.7.55 | 3360 | | 5 | 165 | 21,70 | 2822400 | 840 | 3.360,00 | |
| P.7.56 | 3420 | | 5 | 168 | 22,90 | 2924100 | 855 | 3.420,00 | |

Resultados de estas pruebas son:

| ID Prueba | Nodos del Cluster | Host de carga | Numero de Hilos | periodo entre Hilos | Número de veces | Denegación de servicio |
|-----------|-------------------|---------------|-----------------|---------------------|-----------------|------------------------|
|-----------|-------------------|---------------|-----------------|---------------------|-----------------|------------------------|

Propuesta para la Interoperabilidad de Procesos de Negocio en Sistemas de Información Hospitalarios Bajo una Arquitectura Orientada a Servicios



| | | | | | ejecución | |
|--------|---|---|------|----|------------------|--------|
| P.5.16 | 1 | 3 | 1020 | 25 | 48 | 21,3% |
| P.6.35 | 2 | 4 | 2160 | 5 | 105 | 23,20% |
| P.7.55 | 3 | 5 | 3360 | 5 | 165 | 21,70% |



H GUÍA PARA LA IMPLEMENTACIÓN DE CANAL SOAP EN MITH.

La presente guía está basada en su parte inicial en el anexo B de la tesis para optar a título de Ingeniero en electrónica y Telecomunicaciones de Ricardo Aguilar y Amanecer Urrutia (Aguilar Bolaños, y otros, 2009). Luego es extendida para lograr la construcción de un canal SOAP.

H.1. DESCRIPCIÓN DE MIRTH.

Mirth es una gateway de interfaces o canales HL7 (HL7 channels) que permite el enrutamiento, filtrado y transformación de mensajes. Mirth emplea canales como modelos conceptuales para crear interfaces HL7. El modelo conceptual de un canal se ilustra en la siguiente Figura. En un extremo del canal se tiene el conector de entrada (Source Connector) que se encarga de escuchar, leer o recibir mensajes y en el otro está el conector de salida (Destination Connector) que permite la conexión con diversos sistemas para enviar o escribir mensajes. Cada uno de estos conectores puede configurarse de acuerdo a las necesidades de comunicación, por ejemplo, en esta práctica, el Source Connector permitirá la conexión a una base de datos y la realización de consultas, mientras que el Destination Connector permitirá escribir un archivo que contendrá un mensaje HL7 Versión 3.

Entre ambos extremos del canal es posible tener transformadores (Transformers) y además contar con filtros y/o validadores. Un transformador ejecuta pasos de transformación (Transformer Steps) con los cuales se realiza el “mapeo” de información.

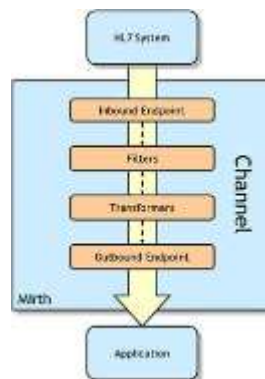


Figura H.1 Modelo conceptual de un canal o interfaz HL7 en Mirth.

H.2. ARRANQUE DEL SERVIDOR (MIRTH SERVER) Y DEL ADMINISTRADOR (MIRTH ADMINISTRATOR).

En Windows 7, en el menú Inicio seleccione Todos los programas, vaya a Mirth y haga clic en Mirth Server. Ahora ejecute el administrador Mirth, para ello, nuevamente desde el menú Inicio, seleccione Todos los programas, vaya a Mirth y haga clic en Mirth Server Manager. En el Área de Notificación de Windows XP (así se denomina la parte inferior derecha de la pantalla justo al lado del reloj) aparecerá un icono con el logo de Mirth Project indicando que el



Mirth Server Manager está disponible; esto se ilustra en la Figura H.2. Con el ratón, al hacer clic derecho sobre este icono, se despliega un menú de opciones, seleccione Launch Administrator.

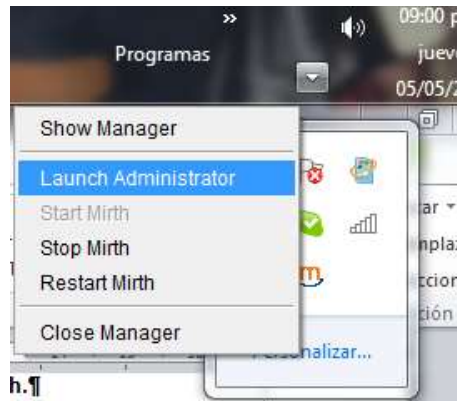


Figura H.2. Icono y menú de opciones de Mirth Server Manager en el Área de Notificación de Windows.

La carga del administrador se inicia (Figura H.3)



Figura H.3. Mirth Connect Manager.

Dentro del proceso de carga aparecerán un par de ventanas con advertencias sobre certificados de seguridad. Es necesario hacer clic en el botón Ejecutar.

La Figura H.4 muestra la ventana que permite validar el acceso e ingresar al sistema. Escriba *admin* como nombre de usuario y contraseña (estos son los valores por defecto).



Figura H.4: Validación Mirth Administrator.

De esta manera se tiene acceso a la interfaz de usuario “Dashboard” o ventana principal. En la parte lateral izquierda se pueden identificar 3 menús: Mirth Connect, Status Tasks y Other.

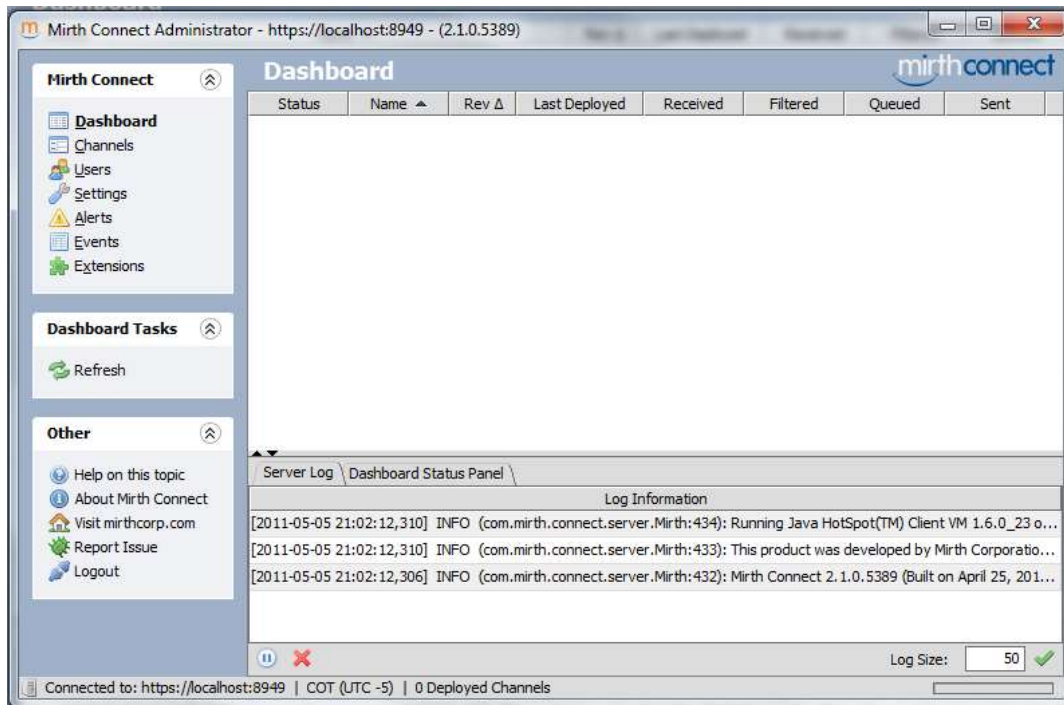


Figura H.5. Ventana principal de Mirth (Dashboard).

H.3 CREACIÓN DEL CANAL Y CONFIGURACIÓN DEL CONECTOR DE ENTRADA (SOURCE CONNECTOR).

En el menú Mirth Connect, haga clic en Channels, aparecerá el menú Channel Task desde el cual al hacer clic en New Channel se despliega la interfaz Edit Channel en la cual se observan cuatro fichas o tabs: Summary, Source, Destinations y Scripts tal como se aprecia en la Figura H.6.

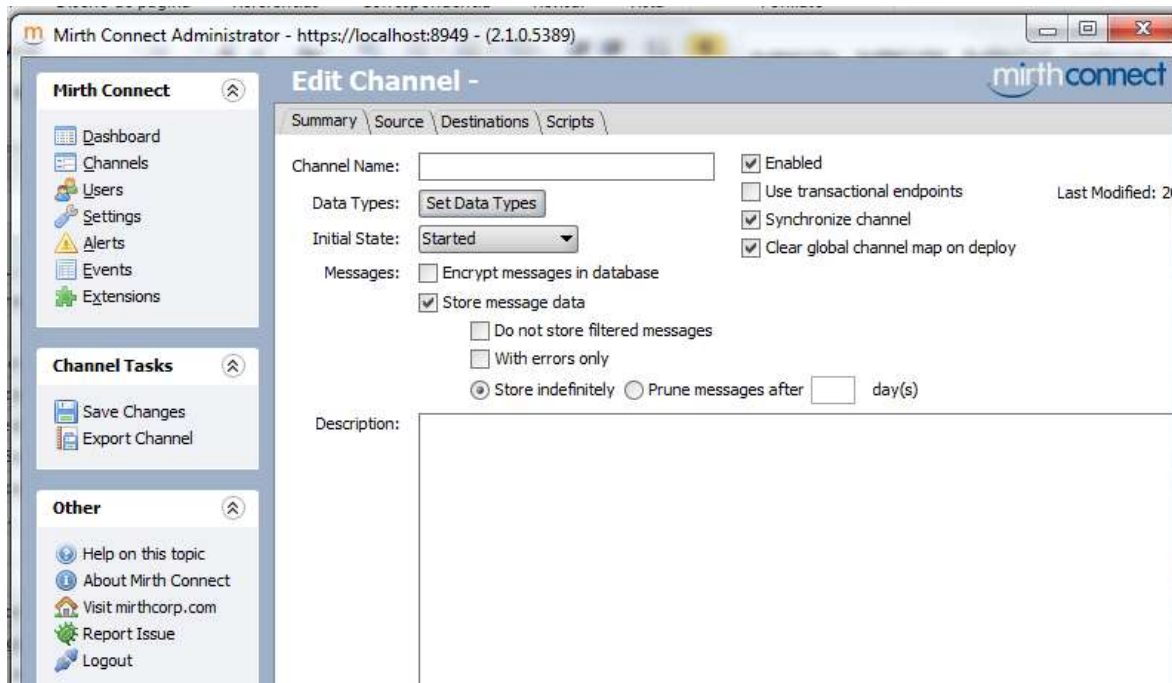


Figura H.6. Ventana para editar un canal (Edit Channel)

En la ficha Summary, asigne un nombre al canal en el campo Channel Name, por ejemplo CanalHL7. En Initial State el estado inicial asígnelo como Stopped o Parado. No es necesario indicar el valor para Incoming Data ya que este valor se ajusta de manera automática cuando se configure el Source Connector. Todas las demás opciones se dejan en sus valores por defecto. Opcionalmente en el campo de texto Description, se puede escribir una descripción del canal. Se puede hacer clic en la opción Save Changes del menú Channel Tasks, si se quiere guardar los cambios realizados o seguir configurando el canal y por último guardar. Si se desea guardar los cambios, aparecerá una ventana de error indicando que hay problemas en la validación del canal; esto se debe a que no se han configurado otros parámetros del canal.

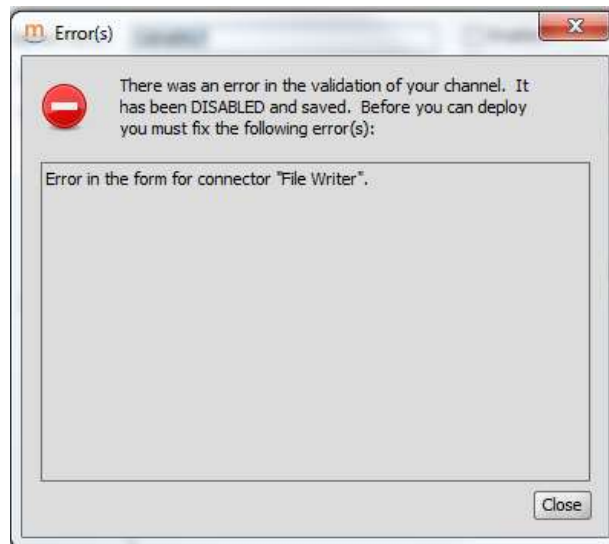


Figura H.7. Ventana de Error.

Ahora, en la ficha Source (Figura B.12) configure el Connector Type como Database Reader. (Otras opciones disponibles son: Channel Reader, FTP Reader, File Reader, HTTP Listener, JMS Reader, JavaScript Reader, LLP Listener, SFTP Reader, SOAP Listener y TCP Listener.)

En el campo Driver especifique el driver que se utilizará para acceder a la base de datos. Mirth cuenta con drivers para: Sun JDBC-ODBC Bridge, MySQL, Oracle 10g Release 2, PostgreSQL y SQL Server/Sybase. En este caso, seleccione MySQL.

En el campo URL escriba la dirección y el nombre de la base de datos a la cual se conectara Mirth. En este caso escriba:

En el campo URL escriba la dirección y el nombre de la base de datos a la cual se conectara Mirth. En este caso escriba:

```
jdbc:postgresql://localhost:5432/soa_sis
```

Si es necesario, especifique información para los campos Username y/o Password.

Ahora, se requiere establecer una sentencia SQL para realizar una consulta en alguna de las tablas de la base de datos. En este caso, en la base de datos existe una tabla denominada "HCE_HISTORIACLINICA"; La sentencia SQL para la consulta es la siguiente:

```
SELECT "ID_HCE_HISTORIA", "ID_PACIENTE", "ID_INGRESO", "MOTIVO_CONSULTA",  
"ANTECEDENTES_ENFERMEDAD", "EXAM_FISICO_OBSERVACION", "EXAM_FISICO_FC",  
"EXAM_FISICO_TA", "EXAM_FISICO_FR", "EXAM_FISICO_PULSO", "EXAM_FISICO_TEMP_AUXILIAR",  
"EXAM_FISICO_TEMP_RECTAL", "EXAM_FISICO_PESO_HABITUAL", "EXAM_FISICO_PESO_ACTUAL",  
"EXAM_FISICO_TALLA", "EXAM_FISICO_BMI", "INSPECCION_GENERAL",  
"PIEL_TEJIDO_CELUALR_SUBCUTANEO", "CABEZA", "CUELLO", "TORAX",  
"APARATO_RESPITARORIO", "APARATO_CARDIOVASCULAR", "ADOMEN_PELVIS",  
"APARATO_GENITOURINARIO", "SISTEMA_NERVIOSO", "SISTEMA_OSTEOMUSCULAR",
```

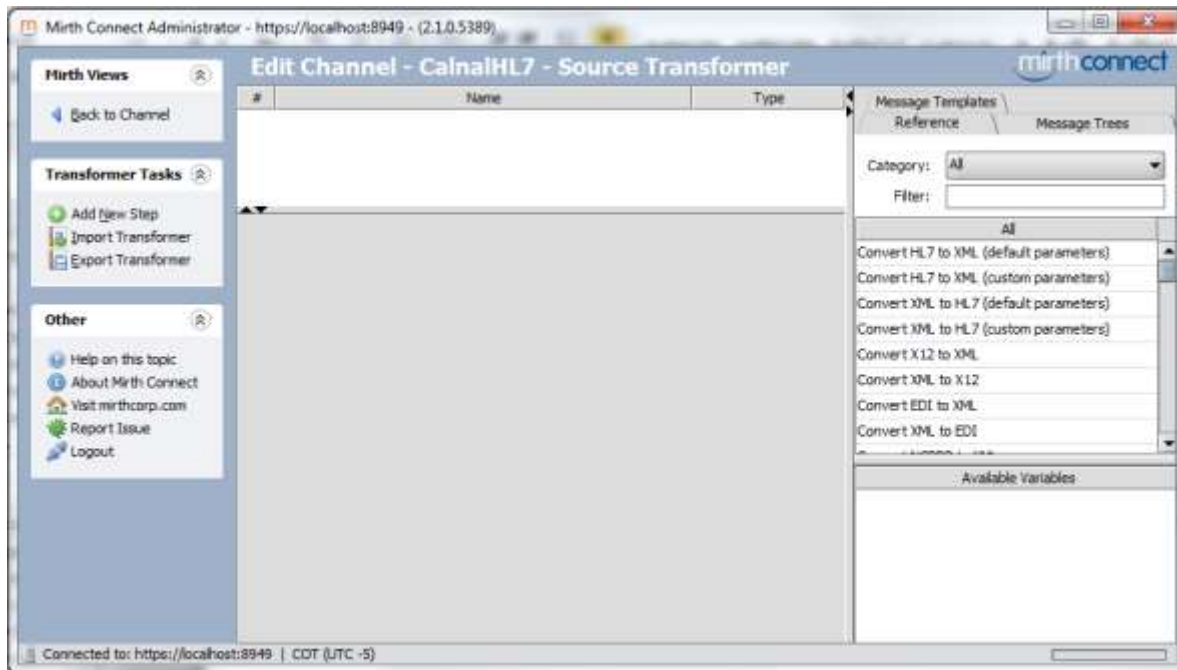



Figura H.9. Ventana de configuración del transformador.

En el menú Channel Tasks, haga clic en la opción Edit Transformer. Esto despliega la interfaz gráfica para Source Transformer que aparece en la Figura H.9. En ella se distinguen dos secciones, además de los menús que aparecen en la parte lateral izquierda: una sección de Pasos de Transformador (Transformer Steps) ubicada en el centro y la otra, ubicada en la parte lateral derecha, compuesta por tres fichas o tabs: Reference, Message Trees y Message Templates.

NOTA. En la ficha Message Templates se definen las plantillas para el mensaje de entrada y el de salida, estas respectivamente son Inbound Message Template y Outbound Message Template. En la plantilla de entrada se define la estructura de la tabla de la base de datos en donde se realiza la consulta.

Seleccione la ficha o tab Message Templates. Esta ficha tiene dos partes: Inbound Message Template y Outbound Message Template (plantillas para mensajes de entrada y salida respectivamente).

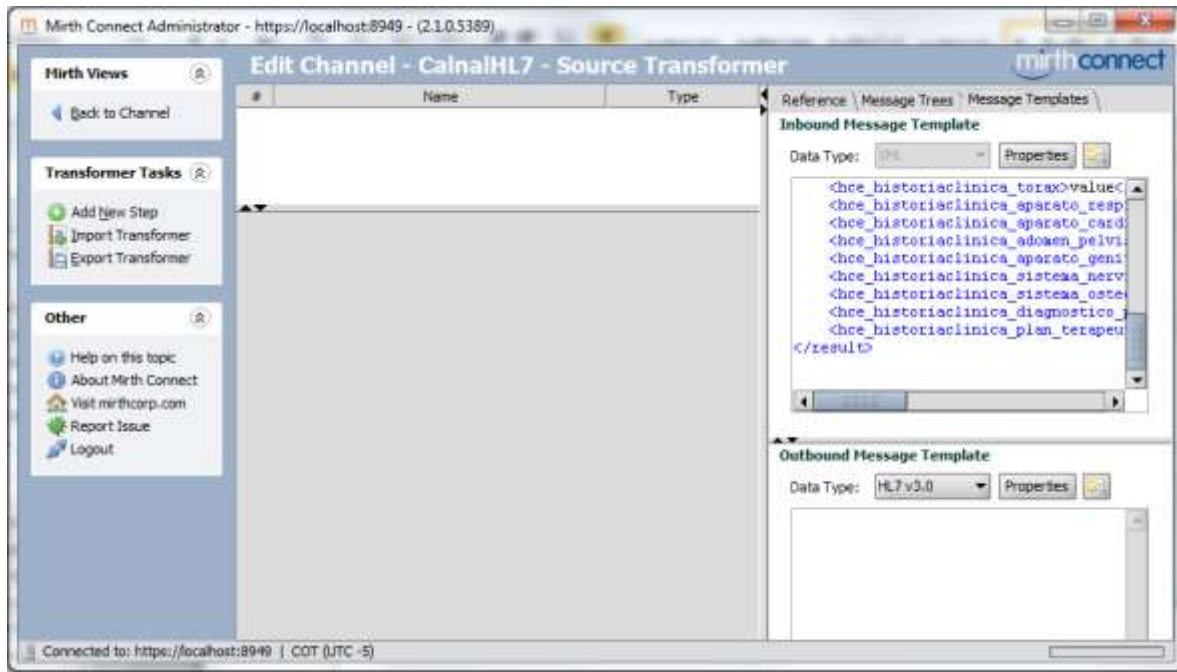


Figura H.10. Ventana de configuración del transformador. Pestaña Message Templates.

En la sección Inbound Message Template se tiene la estructura básica en XML de resultado que se obtiene al consultar una tabla de una base de datos. La Figura H.10 muestra la siguiente información:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<result>
  <hce_historiaclinica_id_hce_historia>value</hce_historiaclinica_id_hce_historia>
  <hce_historiaclinica_id_paciente>value</hce_historiaclinica_id_paciente>
  <hce_historiaclinica_id_ingreso>value</hce_historiaclinica_id_ingreso>
  <hce_historiaclinica_motivo_consulta>value</hce_historiaclinica_motivo_consulta>
  <hce_historiaclinica_antecedentes_enfermedad>value</hce_historiaclinica_antecedentes_enfermedad>
  <hce_historiaclinica_exam_fisico_observacion>value</hce_historiaclinica_exam_fisico_observacion>
  <hce_historiaclinica_exam_fisico_fc>value</hce_historiaclinica_exam_fisico_fc>
  <hce_historiaclinica_exam_fisico_ta>value</hce_historiaclinica_exam_fisico_ta>
  <hce_historiaclinica_exam_fisico_fr>value</hce_historiaclinica_exam_fisico_fr>
  <hce_historiaclinica_exam_fisico_pulso>value</hce_historiaclinica_exam_fisico_pulso>
  <hce_historiaclinica_exam_fisico_temp_auxiliar>value</hce_historiaclinica_exam_fisico_temp_auxiliar>
  <hce_historiaclinica_exam_fisico_temp_rectal>value</hce_historiaclinica_exam_fisico_temp_rectal>
  <hce_historiaclinica_exam_fisico_peso_habitual>value</hce_historiaclinica_exam_fisico_peso_habitual>
  <hce_historiaclinica_exam_fisico_peso_actual>value</hce_historiaclinica_exam_fisico_peso_actual>
  <hce_historiaclinica_exam_fisico_talla>value</hce_historiaclinica_exam_fisico_talla>
  <hce_historiaclinica_exam_fisico_bmi>value</hce_historiaclinica_exam_fisico_bmi>
  <hce_historiaclinica_inspeccion_general>value</hce_historiaclinica_inspeccion_general>
  <hce_historiaclinica_piel_tejido_celualr_subcutaneo>value</hce_historiaclinica_piel_tejido_celualr_subcutaneo>
  <hce_historiaclinica_cabeza>value</hce_historiaclinica_cabeza>
  <hce_historiaclinica_cuello>value</hce_historiaclinica_cuello>
  <hce_historiaclinica_torax>value</hce_historiaclinica_torax>
  <hce_historiaclinica_aparato_respiratorio>value</hce_historiaclinica_aparato_respiratorio>

```



```
<hce_historiaclinica_aparato_cardiovascular>value</hce_historiaclinica_aparato_cardiovascular>
<hce_historiaclinica_adomen_pelvis>value</hce_historiaclinica_adomen_pelvis>
<hce_historiaclinica_aparato_genitourinario>value</hce_historiaclinica_aparato_genitourinario>
<hce_historiaclinica_sistema_nervioso>value</hce_historiaclinica_sistema_nervioso>
<hce_historiaclinica_sistema_osteomuscular>value</hce_historiaclinica_sistema_osteomuscular>
<hce_historiaclinica_diagnostico_presuntivo>value</hce_historiaclinica_diagnostico_presuntivo>
<hce_historiaclinica_plan_terapeutico>value</hce_historiaclinica_plan_terapeutico>
</result>
```

Modifique esta plantilla de tal forma que pueda identificar los campos de la tabla de la base datos. Por ejemplo, de la siguiente manera:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<encounter>
  <hce_historiaclinica_id_hce_historia/>
  <hce_historiaclinica_id_paciente/>
  <hce_historiaclinica_id_ingreso/>
  <hce_historiaclinica_motivo_consulta/>
  <hce_historiaclinica_antecedentes_enfermedad/>
  <hce_historiaclinica_exam_fisico_observacion/>
  <hce_historiaclinica_exam_fisico_fc/>
  <hce_historiaclinica_exam_fisico_ta/>
  <hce_historiaclinica_exam_fisico_fr/>
  <hce_historiaclinica_exam_fisico_pulso/>
  <hce_historiaclinica_exam_fisico_temp_auxiliar/>
  <hce_historiaclinica_exam_fisico_temp_rectal/>
  <hce_historiaclinica_exam_fisico_peso_habitual/>
  <hce_historiaclinica_exam_fisico_peso_actual/>
  <hce_historiaclinica_exam_fisico_talla/>
  <hce_historiaclinica_exam_fisico_bmi/>
  <hce_historiaclinica_inspeccion_general/>
  <hce_historiaclinica_piel_tejido_celualr_subcutaneo/>
  <hce_historiaclinica_cabeza/>
  <hce_historiaclinica_cuello/>
  <hce_historiaclinica_torax/>
  <hce_historiaclinica_aparato_respiratorio/>
  <hce_historiaclinica_aparato_cardiovascular/>
  <hce_historiaclinica_adomen_pelvis/>
  <hce_historiaclinica_aparato_genitourinario/>
  <hce_historiaclinica_sistema_nervioso/>
  <hce_historiaclinica_sistema_osteomuscular/>
  <hce_historiaclinica_diagnostico_presuntivo/>
  <hce_historiaclinica_plan_terapeutico/>
</encounter>
```

Con los cambios realizados a las plantillas de los mensajes de entrada y de salida, puede cambiar a la vista de árbol para ver su estructura (Figura H.11) y poder asignar variables y



condiciones de mapeo en los diferentes pasos de transformación (Transformer Steps) que se requieran. A continuación se describe el procedimiento.

Vaya a la ficha Message Tree. La estructura de los mensajes configurados anteriormente en Ahora cree pasos de transformación (Transformer Steps).

En el menú Transformer Tasks haga clic en la opción Add New Step agregue diferentes pasos de transformación.

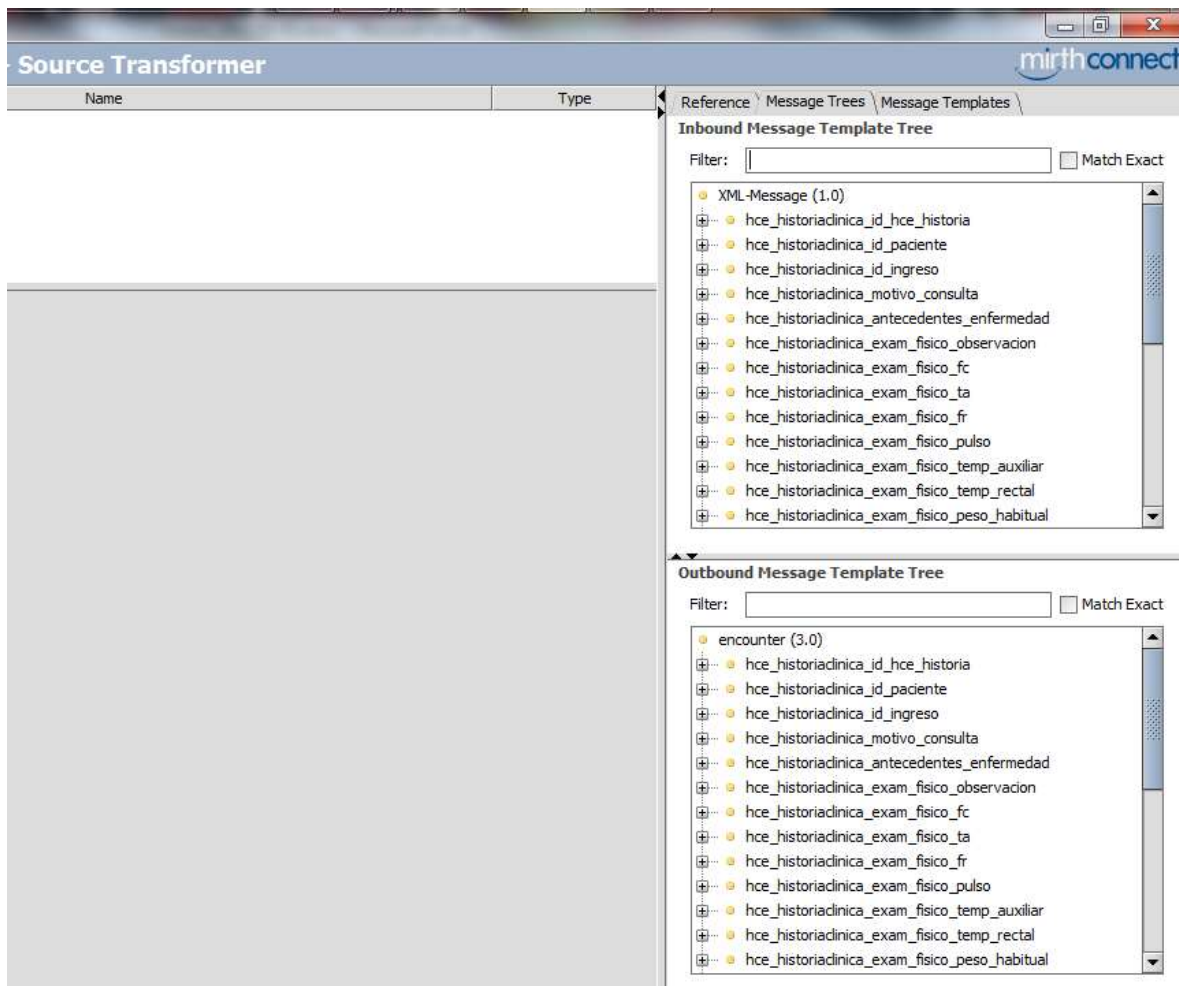


Figura H.11. Ventana de configuración del transformador. Pestaña Message Trees.

Los pasos de transformación incluyen el mapeo de variables y el mapeo de segmentos o la construcción del mensaje. Para la definición de variables de mapeo haga clic derecho en cualquiera de los elementos del árbol del mensaje de entrada (Inbound Message Template Tree) y seleccione la opción Map to Variable. La Figura H.12 ilustra este procedimiento.

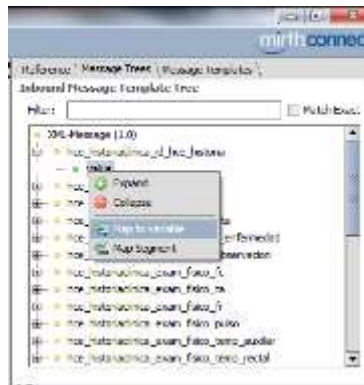


Figura H.12. Especificación de las variables de mapeo en los pasos de transformación

La Figura B.13 muestra siete pasos de transformación (0-28) que se han creado hasta el momento.

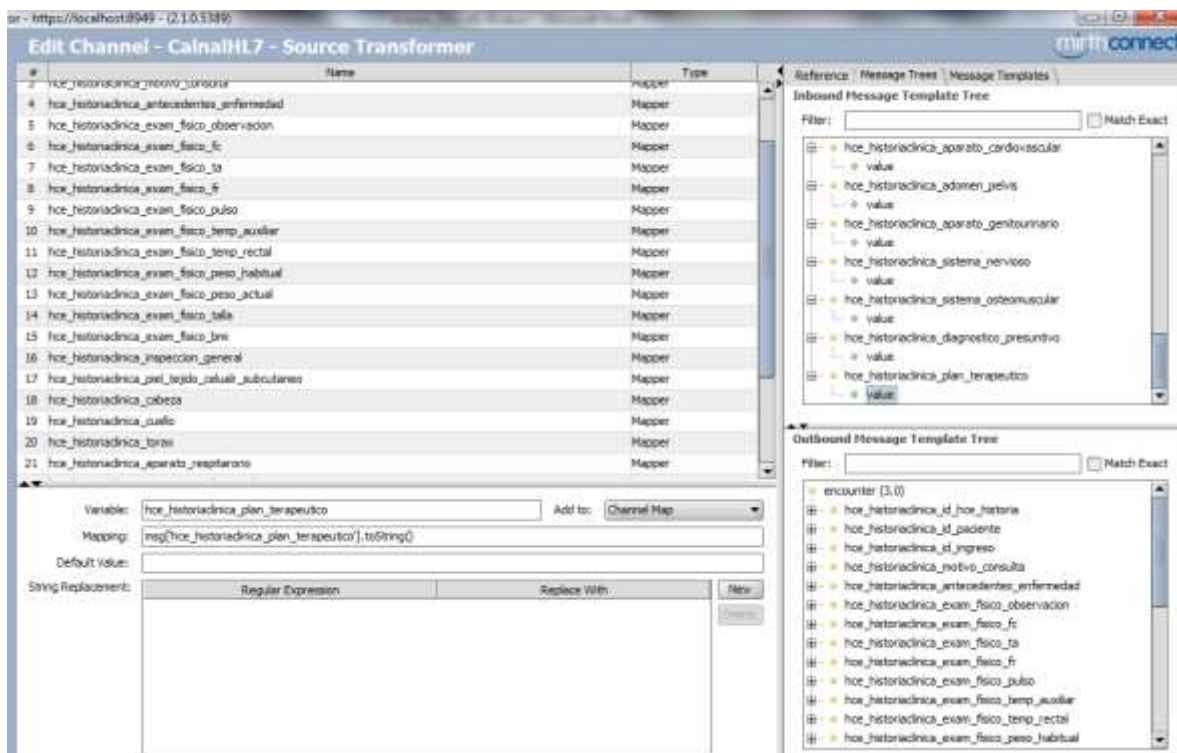


Figura H.13. Ventana de configuración del transformador. Mapeo de variables en los pasos de Transformador.

Los pasos de transformación que involucran el mapeo de segmentos se crean de manera similar, pero utilizando los elementos del árbol del mensaje de salida (Outbound Message Template Tree). Seleccione los elementos de interés, haga clic derecho y elija la opción Map Segment, tal como se ilustra en la Figura H.14.

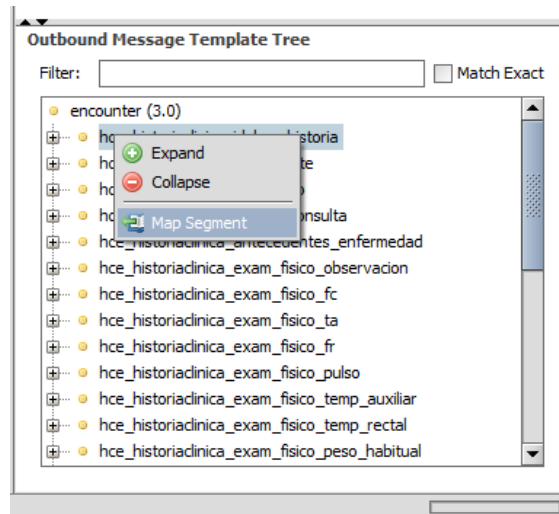


Figura H.14. Especificación del mapeo de segmentos en los pasos de transformación.

La Figura H.16 ilustra los pasos de transformación correspondientes al mapeo de segmentos (pasos 29 al 57).

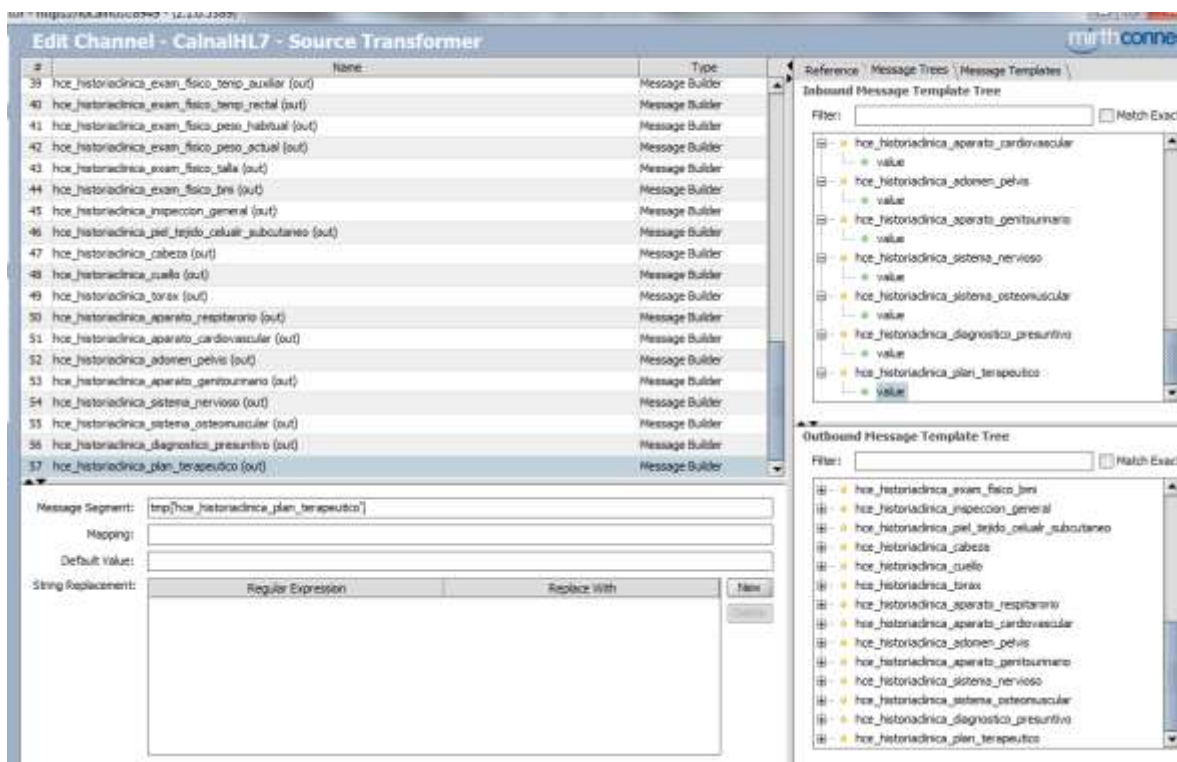


Figura H.15. Ventana de configuración del transformador. Mapeo de segmentos de mensaje en los pasos de transformación (Transformer Steps).

Finalmente, el mapeo se realiza de la siguiente manera: expanda cada elemento del Inbound Message Template Tree y arrastre el elemento indicado con la etiqueta [empty] hasta el campo Mapping del Message Segment correspondiente. La Figura H-16 indica el mapeo para el segmento final id de diagnostico en el paso de transformación número 39.

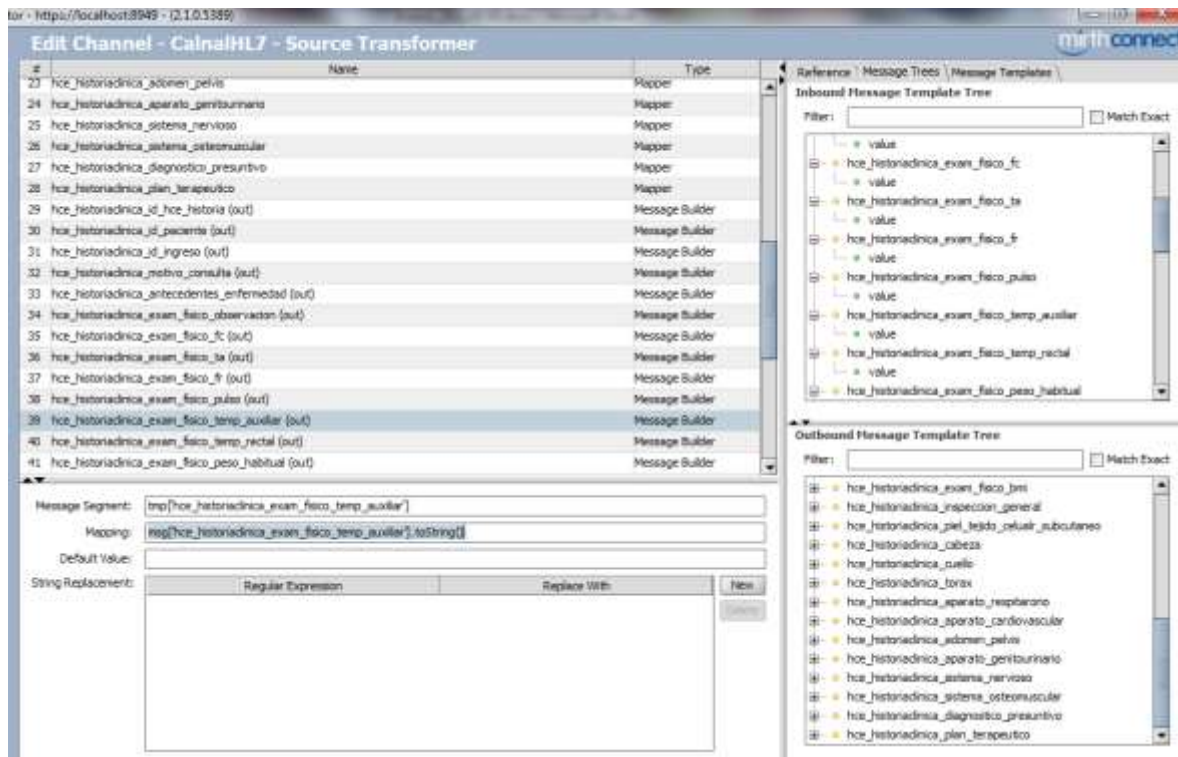


Figura H.16. Mapeo de los segmentos de mensaje.

Luego de haber creado los pasos de transformación, necesarios para el mapeo de la información de la base de datos y la generación del mensaje HL7, haga clic en la opción Back To Channel del menú Mirth Views y para guardar los cambios haga clic en Save Changes del menú Channel Tasks.

Es posible que cuando se estén guardando los cambios aparezca un mensaje de error como el de la Figura H.8, debido a que aún no se ha configurado el Channel Destination. Ésta configuración se realiza a continuación en la siguiente sección.

H.5 CREACIÓN Y CONFIGURACIÓN DEL EXTREMO DESTINO DEL CANAL (CHANNEL DESTINATION).

Haga clic en la pestaña Destinations (Figura H.17). En esta pestaña configure el conector de salida del canal; en los pasos anteriores se configuró el conector de entrada (Channel Source). Para crear un conector de destino puede hacer clic en la opción New Destination del menú Channel Tasks o utilizar el que aparece por defecto: Destination 1

Configure el conector de salida del canal de la siguiente manera: Establezca el tipo de conector en el campo Connector Type seleccionando File Writer. Luego, para este tipo de conector File Writer especifique un directorio (Directory) y un nombre de archivo (File Name). Las demás opciones se dejan por defecto. La Figura H.17 ilustra este procedimiento.



Figura H.17. Ventana de configuración del tipo de conector Database Writer.

En la parte lateral derecha de la ficha Destinations (Figura H.17) aparecen los mapeos de destino (Destination Mappings) y allí se tienen las variables creadas en los pasos de transformación (Transformer Steps). Puede seleccionarlas con el ratón y arrastrarlas a la plantilla (Template).

La configuración final del conector Channel Destination se ilustra en la Figura H.17. Finalmente, para guardar la configuración del canal, vaya al menú Channel Tasks y seleccione la opción Save Changes.

H.6 PUBLICACIÓN Y DESPLIEGUE DEL CANAL

Haga clic en la opción Channels del menú Mirth. El canal debe aparecer en la lista de canales disponibles y su estado (status) debe ser Enabled (Ver Figura H.18).

Ahora en el menú Channel Tasks seleccione la opción Deploy All. El estado del canal pasa a Started y se puede ver información relacionada con la conexión y estado del canal en la sección Dashboard Status Panel. Así mismo, es posible observar la actividad en la consola del servidor Mirth Server.

El canal puede ser arrancado y parado seleccionando del menú Status Tasks las opciones Start Channel o Stop Channels. El canal también puede ser reseteado o puesto en pausa mediante las opciones Reset All Channels o Pause Channel. Estas opciones también están disponibles al hacer clic derecho sobre el canal de interés.

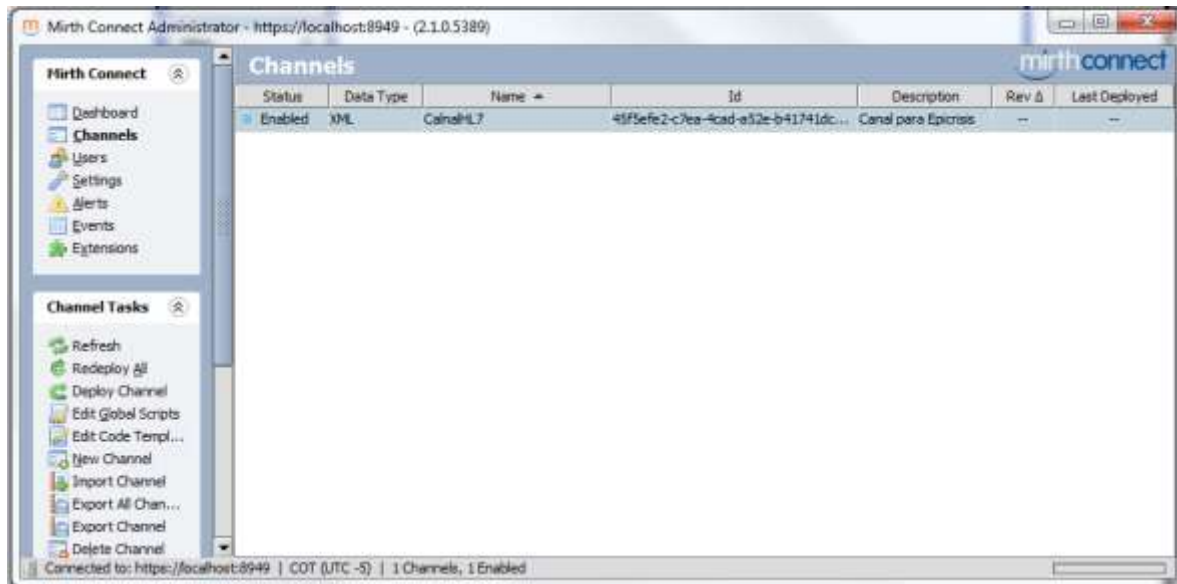


Figura H.18. Ventana para publicación y despliegue del canal.

H.6 PUBLICACIÓN Y DESPLIEGUE DE UN CANAL SOAP.

Un conector SOAP crea un servicio web que escucha y puede ser invocado para enviar mensajes al canal previamente creado.

Se configuran atributos básicos como el nombre de dominio DNS o la dirección IP en la que el servicio web, debe escuchar las conexiones, el puerto en el que el servicio web, debe escuchar las conexiones, el nombre que dan al servicio web. Podemos ver la URL generada WSDL para el servicio web, así el cliente que envía mensajes al servicio pueden descargar este archivo para determinar cómo llamar al servicio web. También el nombre del método. Y el canal de destino que abastecerá un valor de retorno mediante la hoja de respuesta XML.

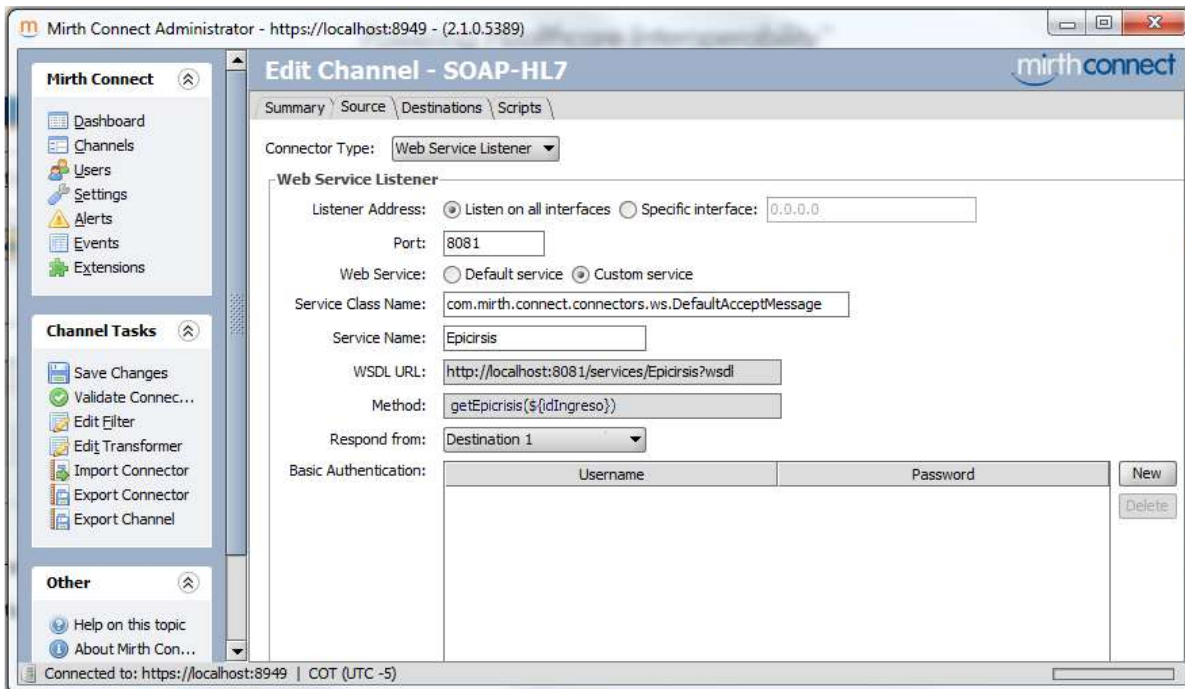


Figura H.19. Crear Canal SOAP.

Posteriormente el método que esta encapsulado para que sea reportando el archivo HL7.

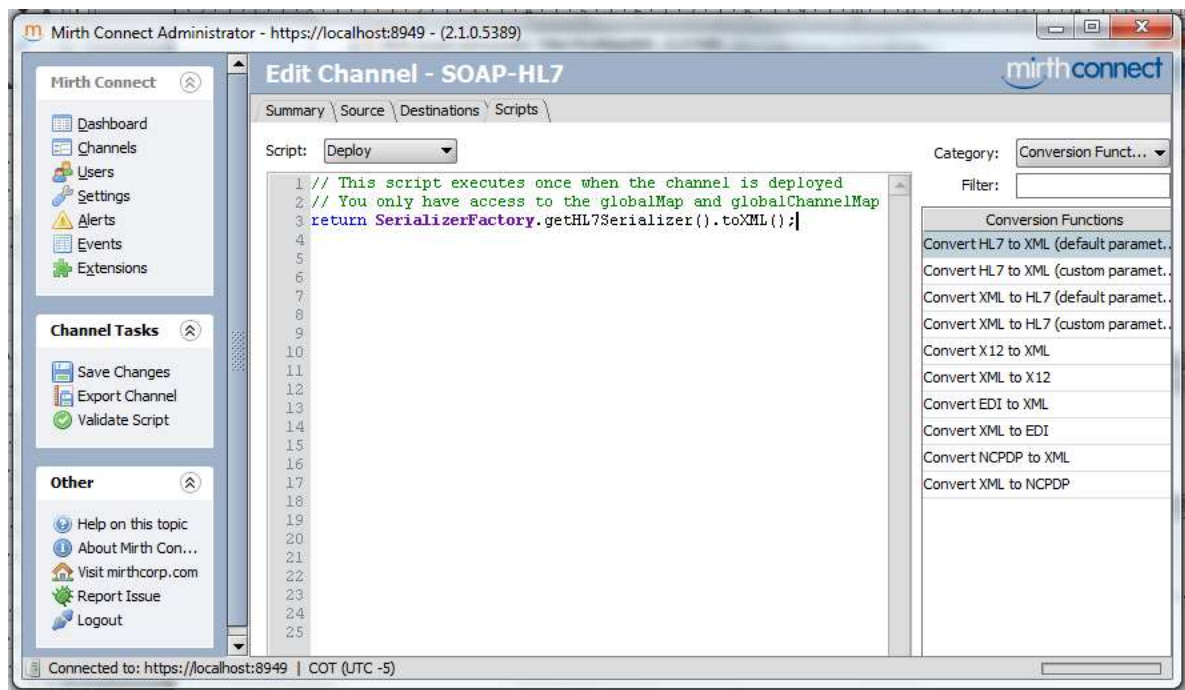


Figura H.20. Metodo encapsulado de respuesta.

Posteriormente desde la pestaña Source se valida el funcionamiento del canal como se muestra en la Figura H.21.

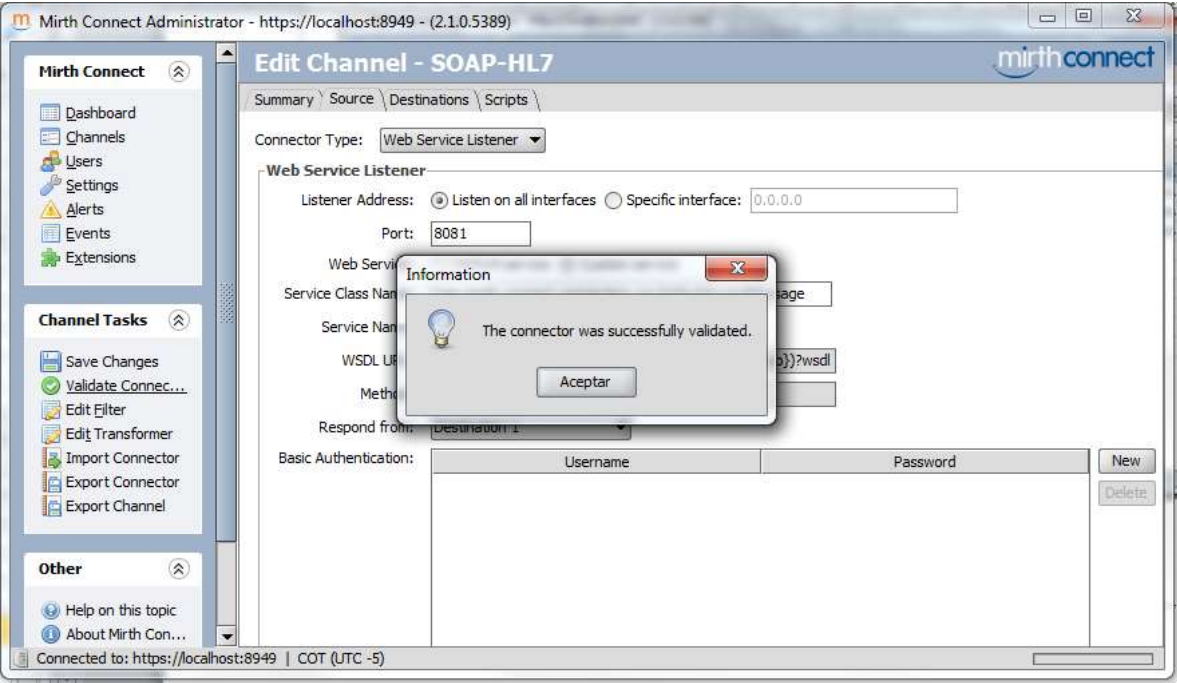


Figura H.21. Validación del canal.



I GUÍA DE IMPLEMENTACIÓN DE UN SERVICIO WEB CON TRASFORMACIÓN HL7 EMPLEANDO CHAINBUILDER (GETTING STARTED WITH CHAINBUILDER CONNECT).





REFERENCIAS

Apache Software Foundation. 2008. Apache Synapse - The lightweight ESB. [En línea] 09 de Junio de 2008. [Citado el: 23 de Marzo de 2010.] <http://synapse.apache.org/index.html>.

Apache Software Foundation. 2010. Apache ServiceMix, the agile Open Source ESB. [En línea] 27 de Abril de 2010. [Citado el: 05 de Septiembre de 2010.] <http://servicemix.apache.org>.

ChainBulder Connect. 2010. ChainBulder Connect. [En línea] Septiembre de 2010. [Citado el: 12 de Agosto de 2010.] <http://www.chainforge.net/cbesbWiki/>.

FUSE Community. 2010. FUSE Open Source Community. [En línea] 2010. [Citado el: 09 de Agosto de 2010.] <http://fusesource.com/>.

HSSP. 2009. The Practical Guide for SOA in Health Care: A real-world approach to planning, designing, and deploying SOA. [En línea] 29 de December de 2009. [Citado el: 3 de November de 2009.] <http://hssp.wikispaces.com/file/view/2008-12-29+SOA%2BHealthcare%2BPractical%2BGuide%2Bv1.0.pdf>.

JBoss. 2010. JBoss ESB - JBoss Community. [En línea] 2010. [Citado el: 02 de Septiembre de 2010.] <http://www.jboss.org/jbossesb>.

Jee(G)ESB. 2010. java.net The Source Java Technology Collaboration. [En línea] 2010. [Citado el: 10 de Septiembre de 2010.] <https://jeeesb.dev.java.net/>.

Mulesoft. 2010. Mule ESB - Open Source Community w. Documentation | Mule ESB Community. [En línea] 2010. [Citado el: 06 de Abril de 2010.]

ObjectWeb. 2008. Celtix: The Open Source Java Enterprise Service Bus. [En línea] 01 de Mayo de 2008. [Citado el: 27 de Febreo de 2010.] <http://celtix.ow2.org/>.

Oracle. 2010. OpenESB. [En línea] 2010. [Citado el: 12 de Enero de 2010.] <https://open-esb.dev.java.net/>.

Petals ESB. 2010. Petlas ESB, The Open Source ESB for large SOA Infraestructures. [En línea] 2010. [Citado el: 18 de Enero de 2010.] <http://petals.ow2.org/>.

WSO2. 2010. Open Source ESB by WSO2 - Ultra fast Open Source Service Bus (ESB) based on Apache Synapse ESB. [En línea] 2010. [Citado el: 25 de Febrero de 2010.] <http://wso2.com/products/enterprise-service-bus/>.