Índice

[1 Introducción a Oracle Service Bus 2](#_Toc468284064)

[1.1 Integración para exponer en el bus un servicio SOAP 3](#_Toc468284065)

[1.1.1 Creación de recursos 9](#_Toc468284066)

[1.1.2 Creación de Proxy: 22](#_Toc468284067)

[1.1.3 Instalación y prueba de servicio integrado al bus. 29](#_Toc468284068)

# Introducción a Oracle Service Bus

Por el surgimiento de la Arquitectura Orientada a Servicios (SOA) y de la necesidad que tienen los clientes de comunicarse con socios y demás. Dentro del contexto de una arquitectura orientada a servicios se destaca el concepto "Bus de servicios", cuyo objetivo es proveer una capa de virtualización de servicios entre los proveedores y los clientes. Si hacemos referencia a la matriz de madurez que provee The Open Group (www.opengroup.org) llamada OSIMM (Open group Service Integration Maturity Model), el Bus de Servicios es el encargado de cumplir con el nivel de madurez llamado "Servicios Virtualizados".

Es muy importante contar con esta capa de virtualización pues de esta manera los clientes no consumirán directamente los servicios expuestos por los proveedores, proporcionando agilidad para poder realizar transformaciones, orquestaciones, cambio en interfaces y desacoplamiento entre los servicios expuestos y/o los servicios consumidores.

**Beneficios de utilizar OSB:**

• Virtualización de Servicios  
• Ruteos   
• Orquestamiento   
• Cambio en interfaces   
• SLA's a nivel de servicios  
• Balanceo entre varios servicios desplegados en distintos servidores  
• Desacoplamiento entre los clientes y los proveedores  
• Aseguramiento de Servicios  
• Agilidad en los cambios de interfaces  
• Migraciones virtuales de servicios  
• Cambios ágiles en las interfaces sin afectar a los clientes

Los componentes elementales del bus de servicios son dos:

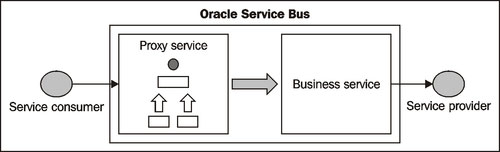
1. Business Service  
2. Proxy Service

**Business Service:** Es el servicio final que expone un determinado proveedor.  
**Proxy Service:** Es un servicio virtual que provee el bus de servicios y en el cual se puede realizar ruteos, orquestaciones y demás.

**Flujo de datos:**

Un cliente consume el servicio virtual expuesto por el bus de servicios, dentro de éste servicio virtual (Proxy Service) se realiza algunas transformaciones y/o ruteos, luego los datos transformados son enviados al servicio final (Business Service), éste procesa la información y nuevamente regresa el resultado al servicio virtual, para después ser entregado al cliente.

En la siguiente imagen se puede observar cómo interactúa el "Business Service" con el "Proxy Service":

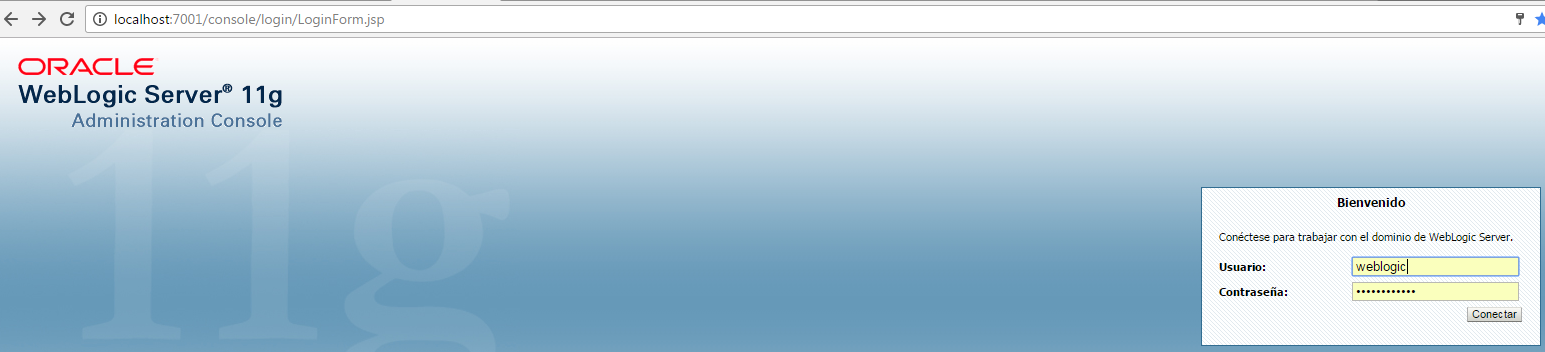


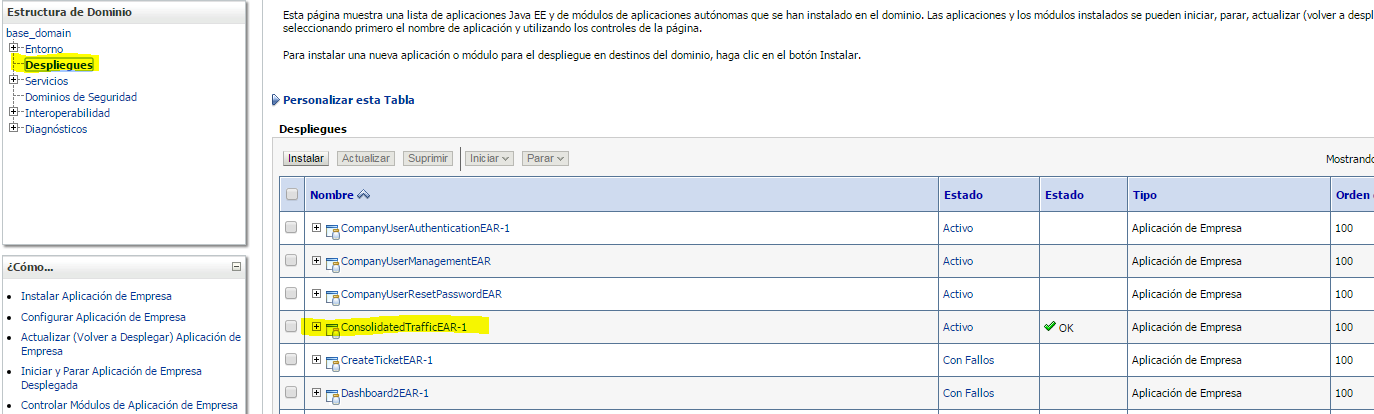
## Integración para exponer en el bus un servicio SOAP

* Primero se necesita tener un Oracle osb instalado con las siguientes características mínimas:  
  [Parte 1: Oracle Database XE 11g](http://uanscarvalho.com.br/instalacao-do-oracle-soasuite-11g-parte-1-oracle-database-xe/)
* [Parte 2: Java SE Development Kit 7 – JDK](http://uanscarvalho.com.br/instalacao-oracle-soasuite-11g-parte-2-java-se-development-kit-7-jdk/)
* [Parte 3: Oracle WebLogic Server 10.3.6](http://uanscarvalho.com.br/instalacao-oracle-soasuite-11g-parte-3-oracle-weblogic-server-10-3-6/)
* [Parte 4: Repository Creation Utility (RCU) 11.1.1.7.0](http://uanscarvalho.com.br/instalacao-oracle-soasuite-11g-parte-4-repository-creation-utility-rcu/)

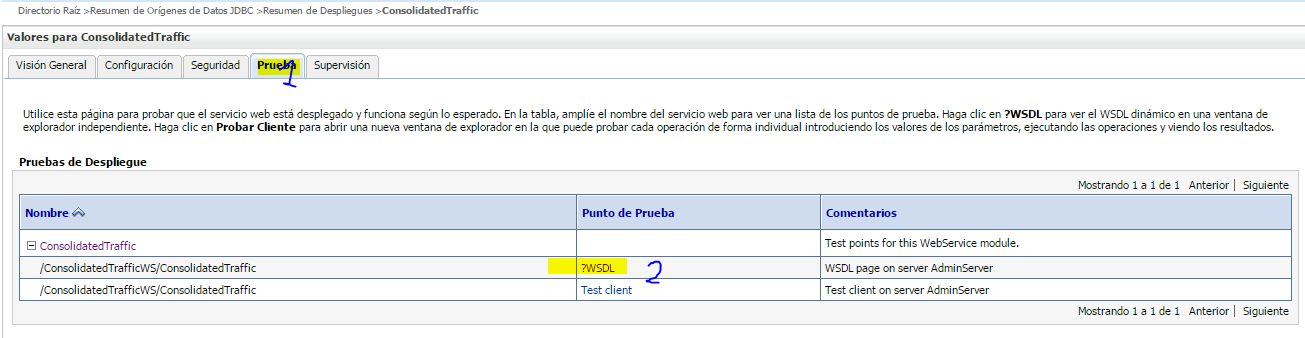
Ahora si se quiere crear proyectos OSB con el ide eclipse se debe instalar:  
**Oracle Enterprise Pack for Eclipse 11gR1 (11.1.1.8)**

[Manual de instalación](http://uanscarvalho.com.br/instalando-e-configurando-oracle-service-bus-11g/?lang=es)Ejemplo de aplicación OSB. **Teniendo ya todo instalado solo necesitamos el WSDL y el XSD del webservice SOAP que queremos exponer, se puede obtener desde la consola en donde publicamos el proyecto SOAP.**



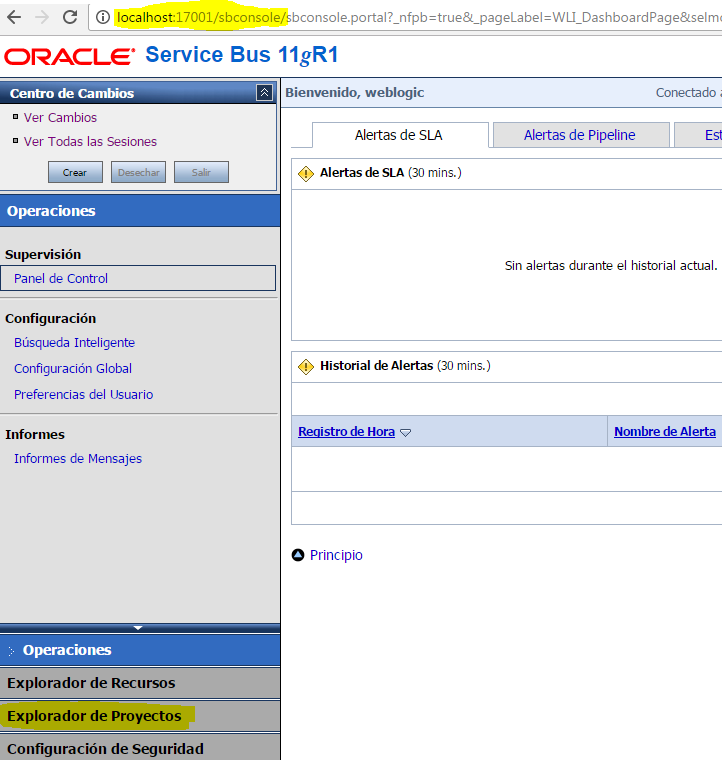
****

****



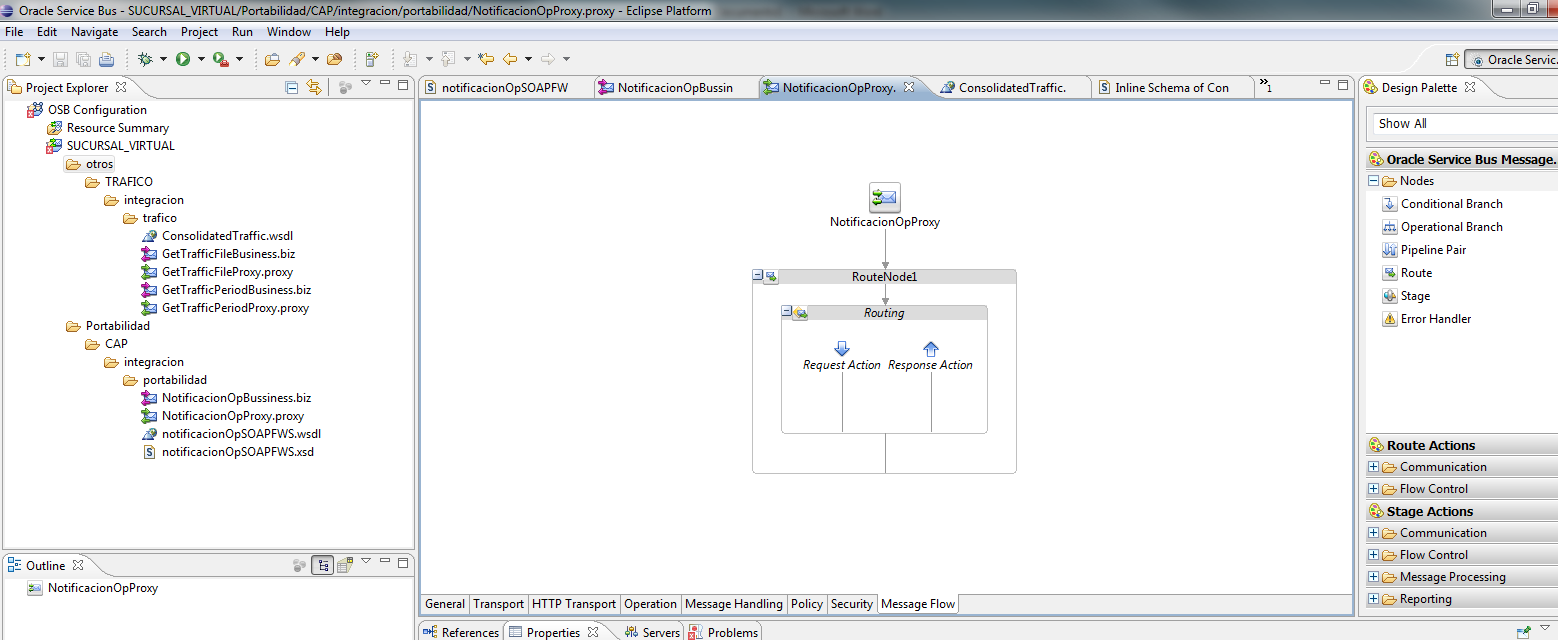


Teniendo el archivo se pueden crear los proyectos de dos formas usando la consola de administración del OSB: IP:PUERTO/sbconsole

Ej: <http://localhost:17001/sbconsole>  
  
  
  
Tiene inconvenientes como timeout de sesión, algunos cambios se demoran en guardar, navegación en muchas pantallas, se pierden cambios entre sesiones si no se es ordenado, las modificaciones sobre un mismo proyecto son complicadas, no hay mantención de los aplicativos, no se puede llevar un respaldo un error y se pueden perder los proyectos por un error humano o error de configuración de servidor.

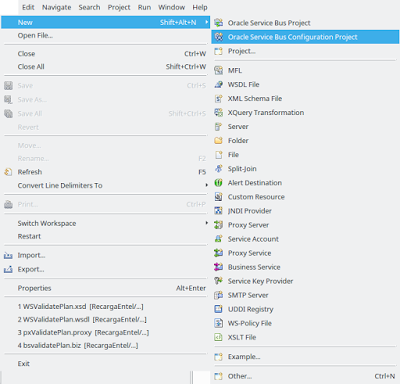
O la otra forma es usando el OEPE:

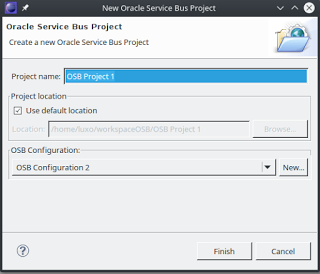


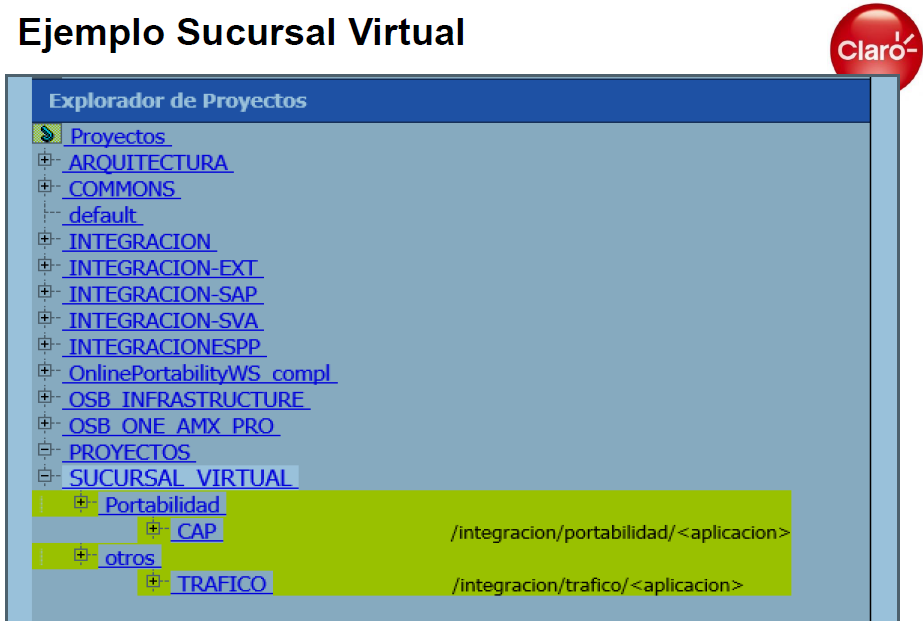


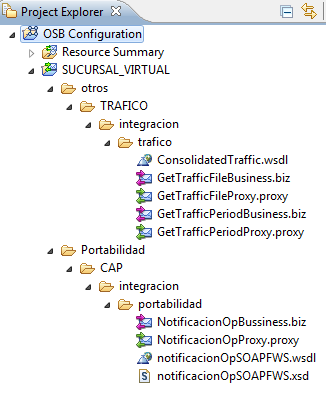
Es un Eclipse para SOA, se puede versionar como otro proyecto más en GIT.

Pasos en eclipse oepe para crear un proyecto OSB:



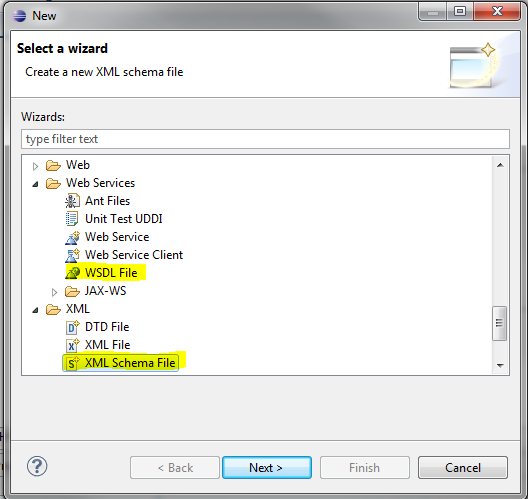


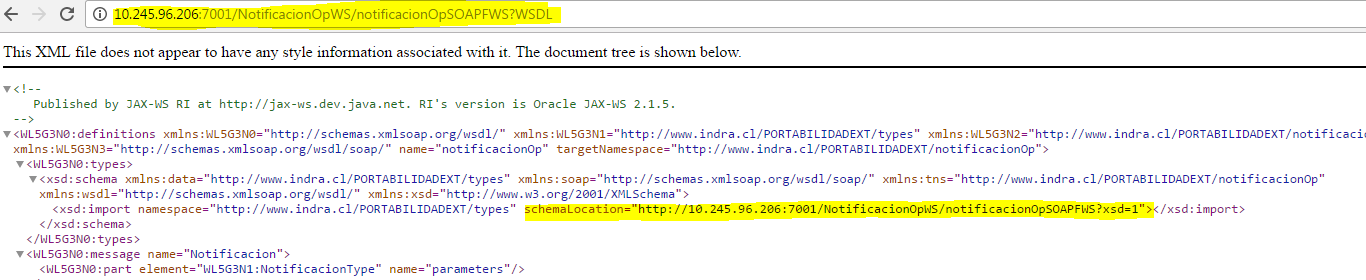
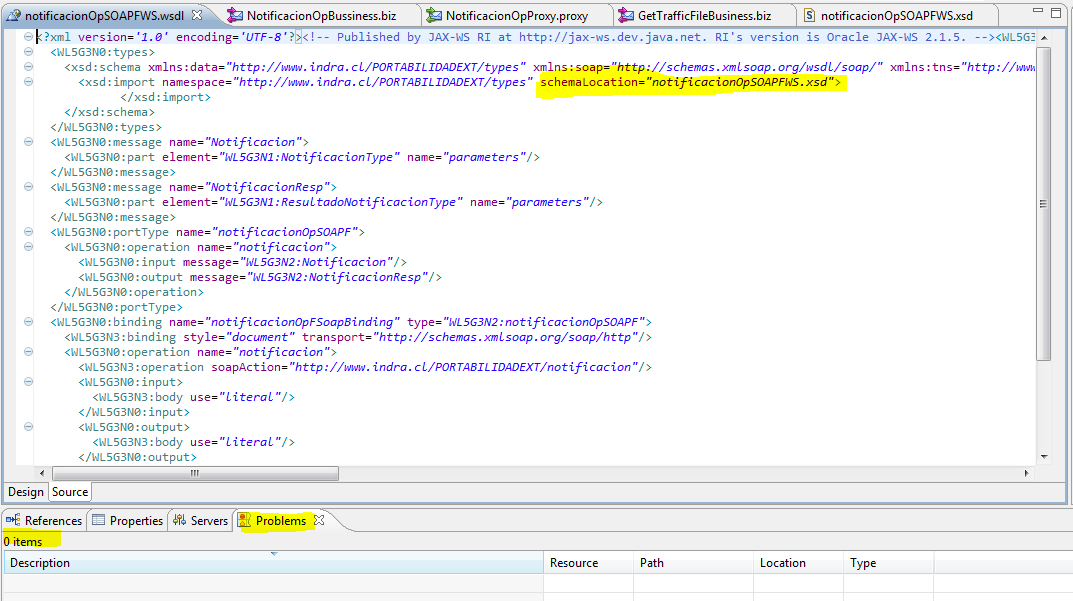
Poniendo como ejemplo la documentación enviada por Claro:  
  
La estructura de la aplicación quedaría como:

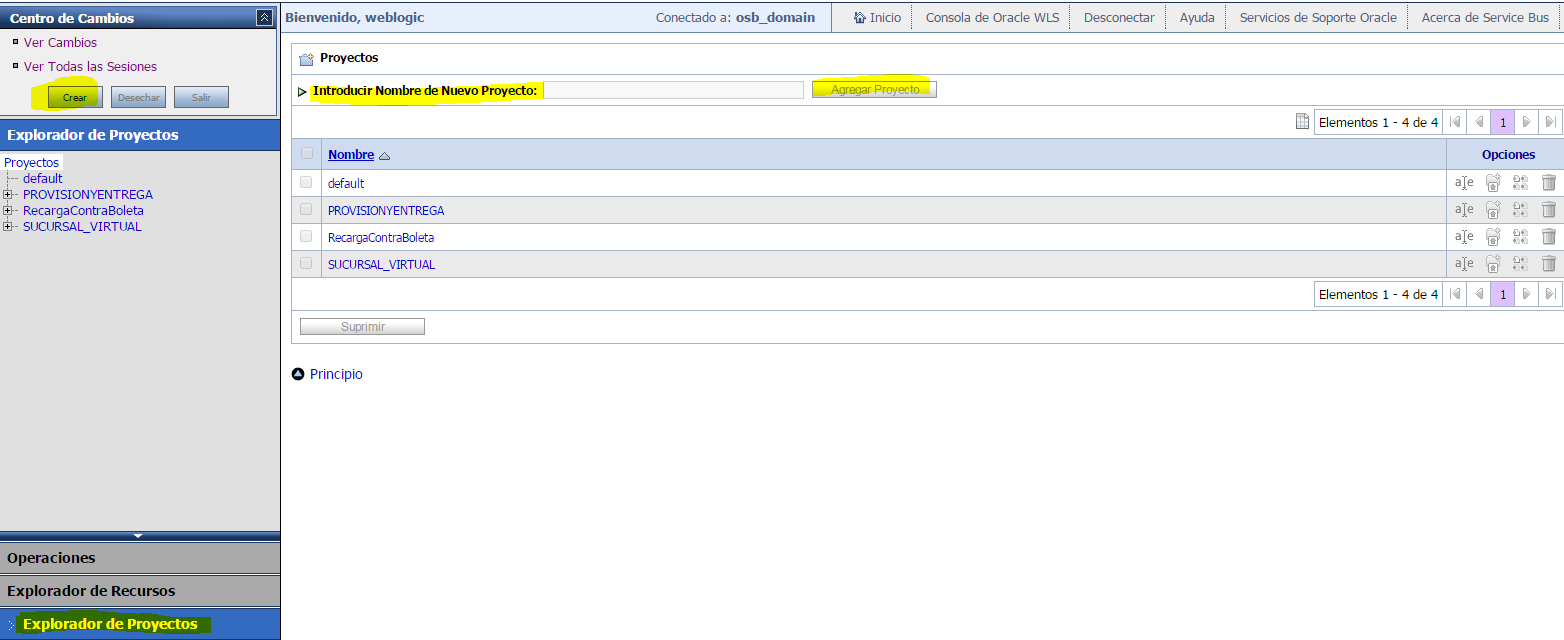


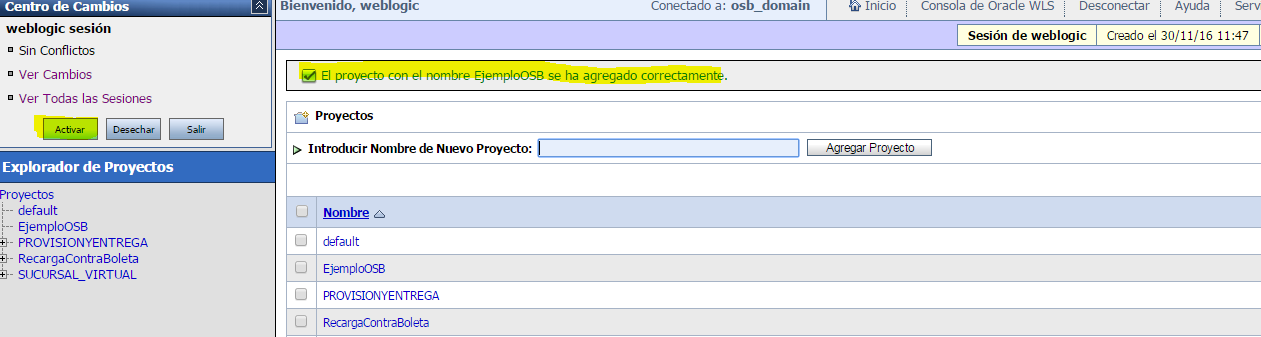
### Creación de recursos

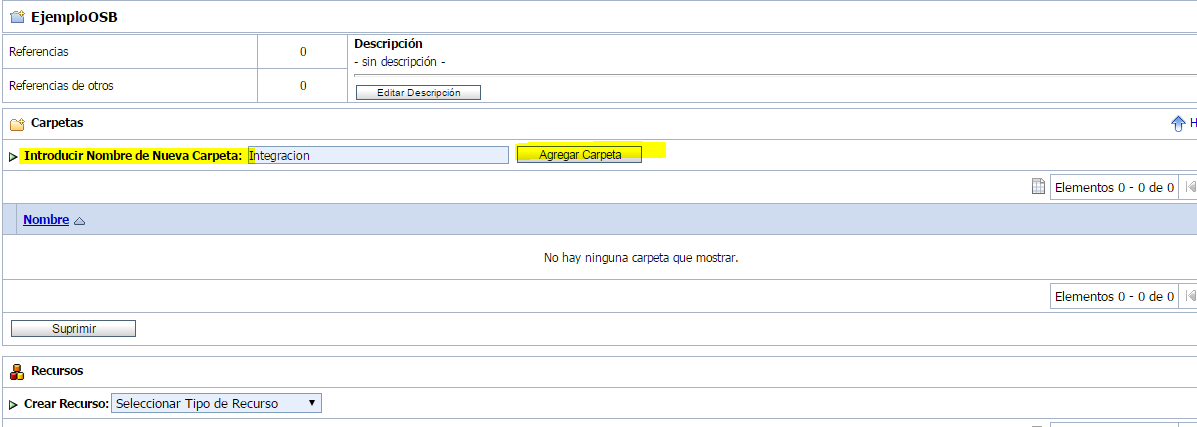
Teniendo la estructura creada para el cliente se crean los resources xsd y wsdl, se importan al proyecto y o se crean los archivos en el IDE.

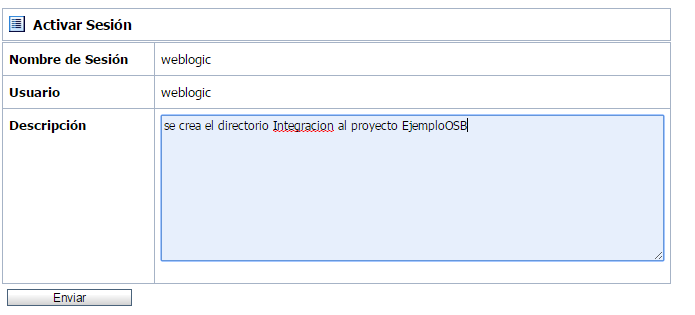


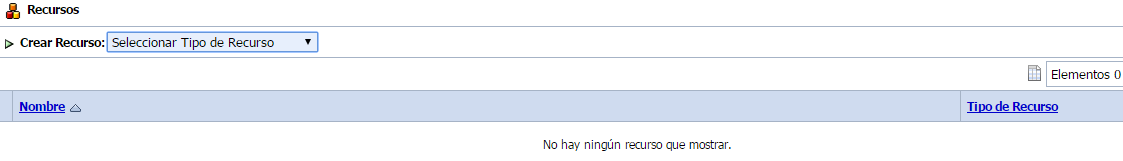
Una vez creados los recursos faltaran editarlos algunas veces para hacer que tengan consistencia las referencias entre wsdl y sus esquemas, lo ideal es tener un wsdl y un solo defincion xsd.  
En el original:  
  
Pero debe editarse a:  
  
Se pueden corregir los problemas en el proyecto como cualquier proyecto en eclipse.

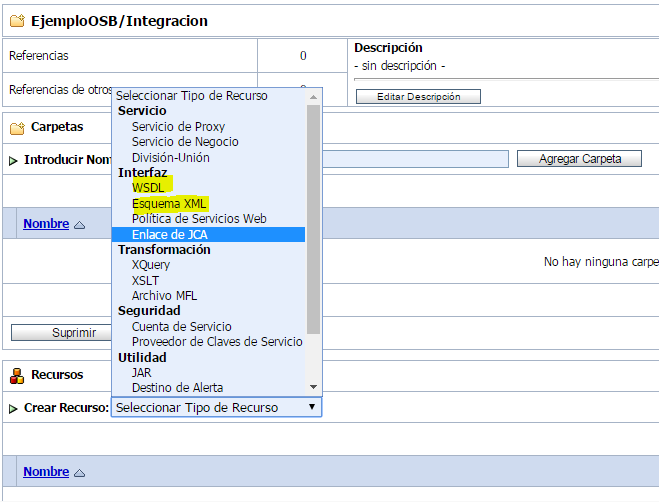
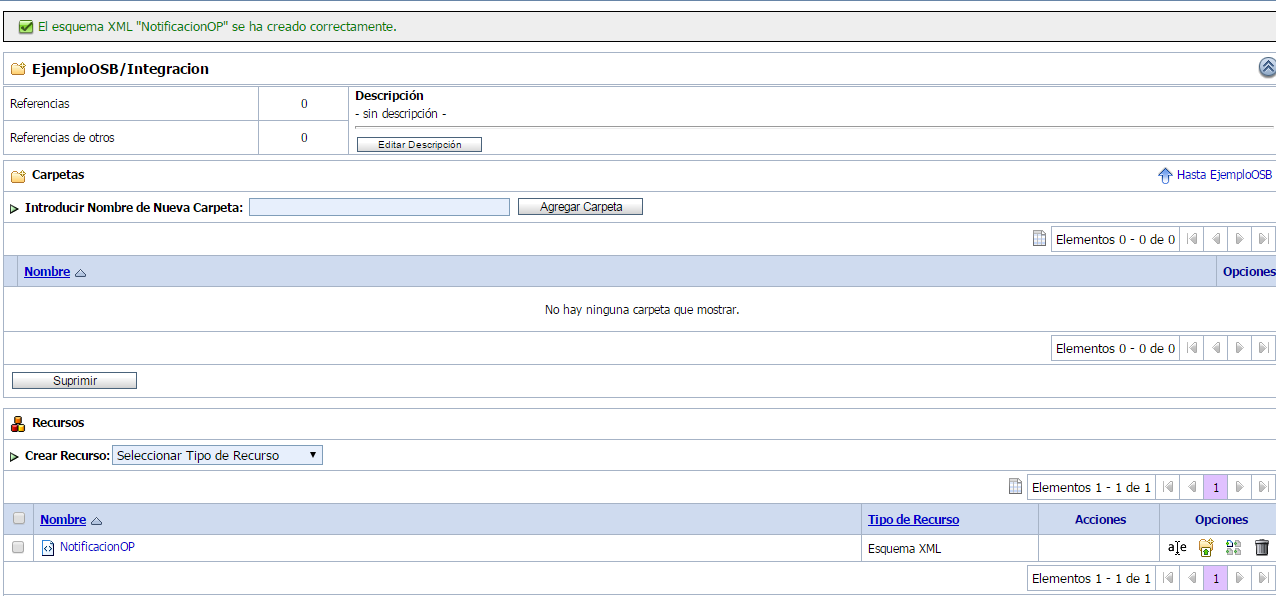
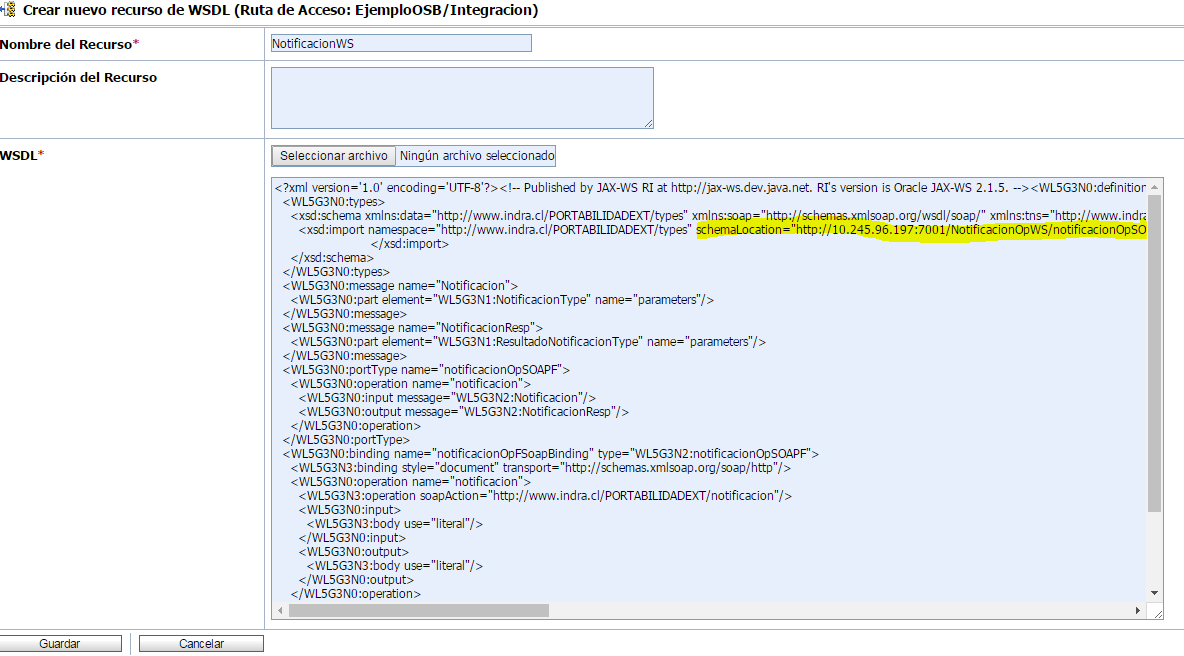
El mismo procedimiento tiene su símil en la consola del OSB:  


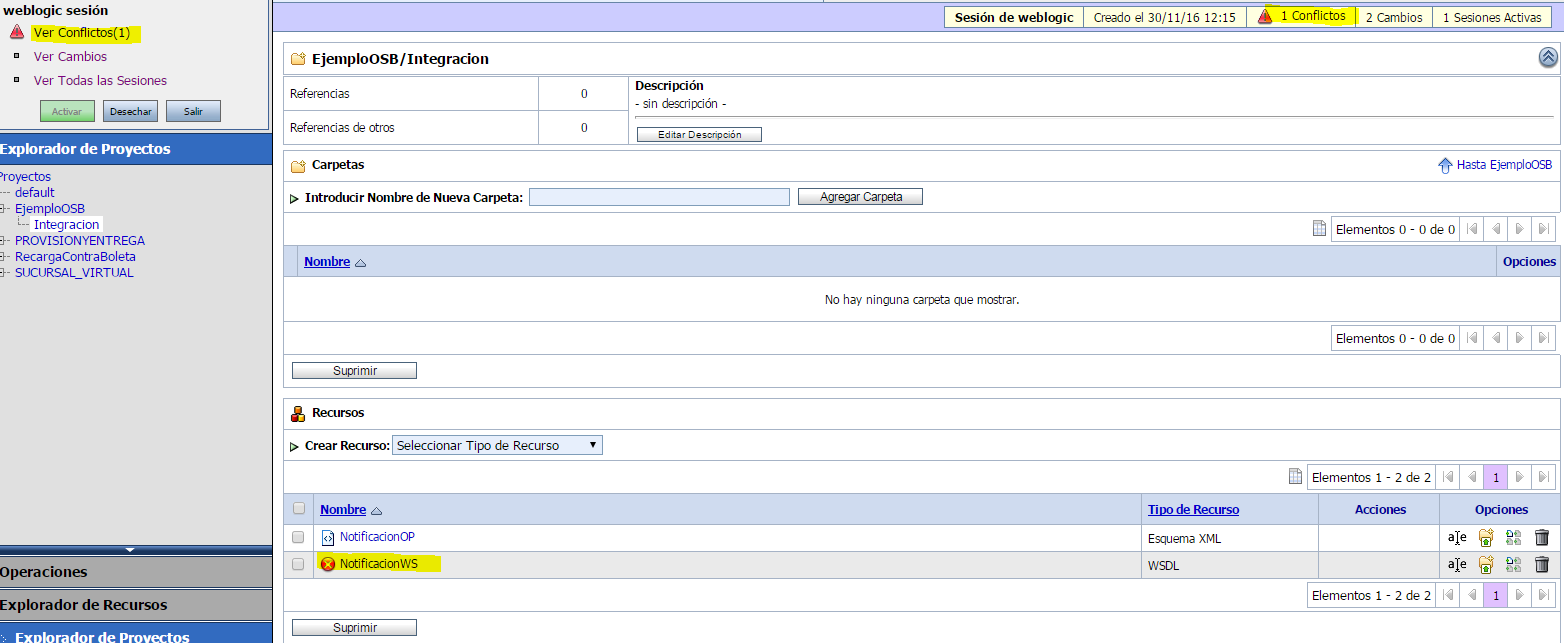
Para cada cambio que se haga se deben activar los cambios, si no se guardan constantemente provocaran conflictos en la sesión, por ej : eliminación de un proxy y en 3 pasos siguientes modificaciones al mismo proxy, todo cambio que se haya hecho a otro modulo queda en conflicto, la única solución es eliminar el cambio y volver atrás perdiendo las modificaciones a los otros módulos.  
  


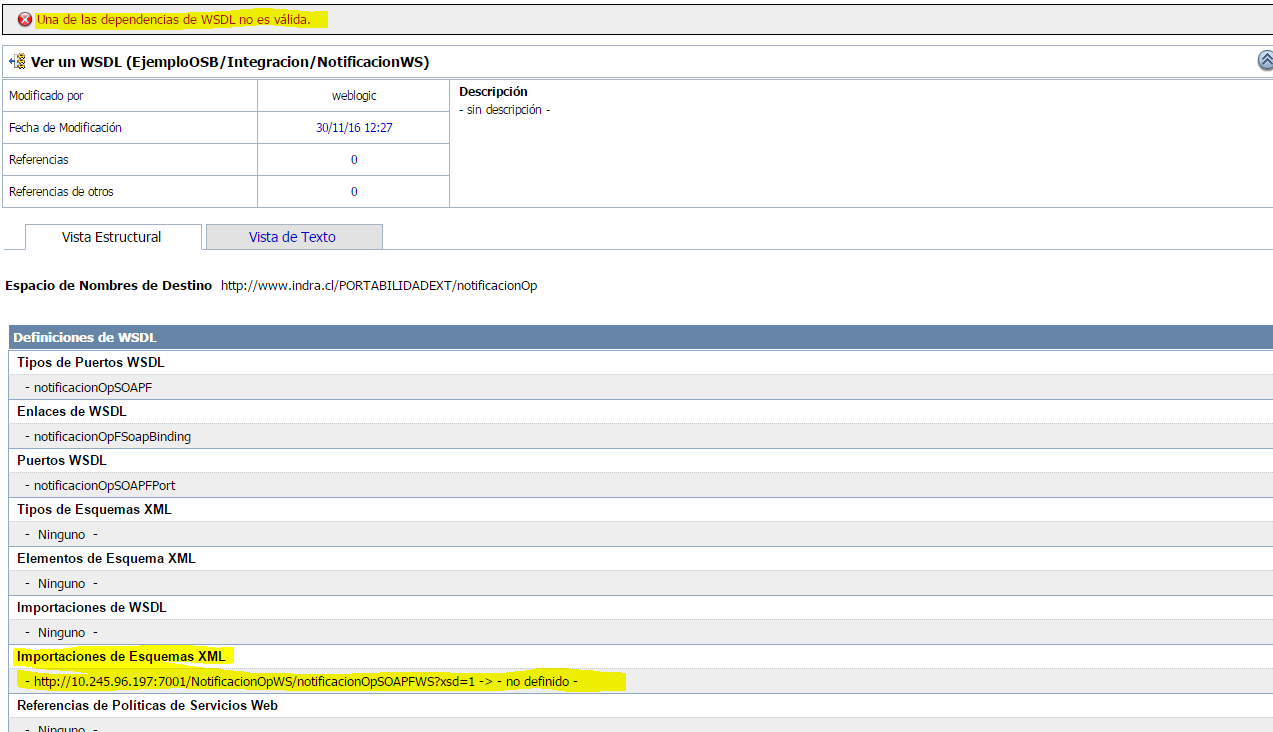


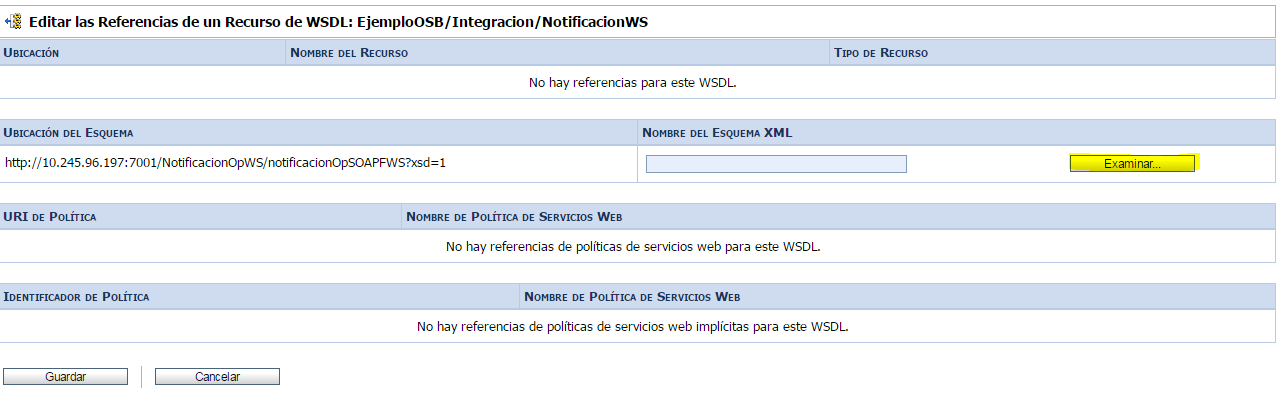
  
  

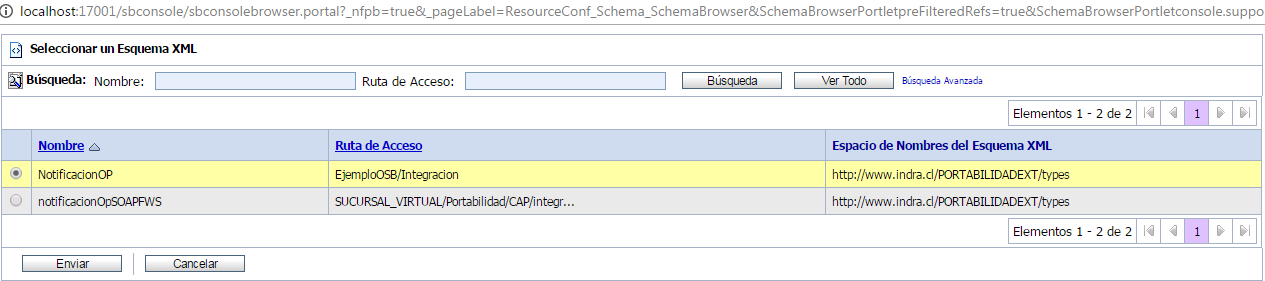



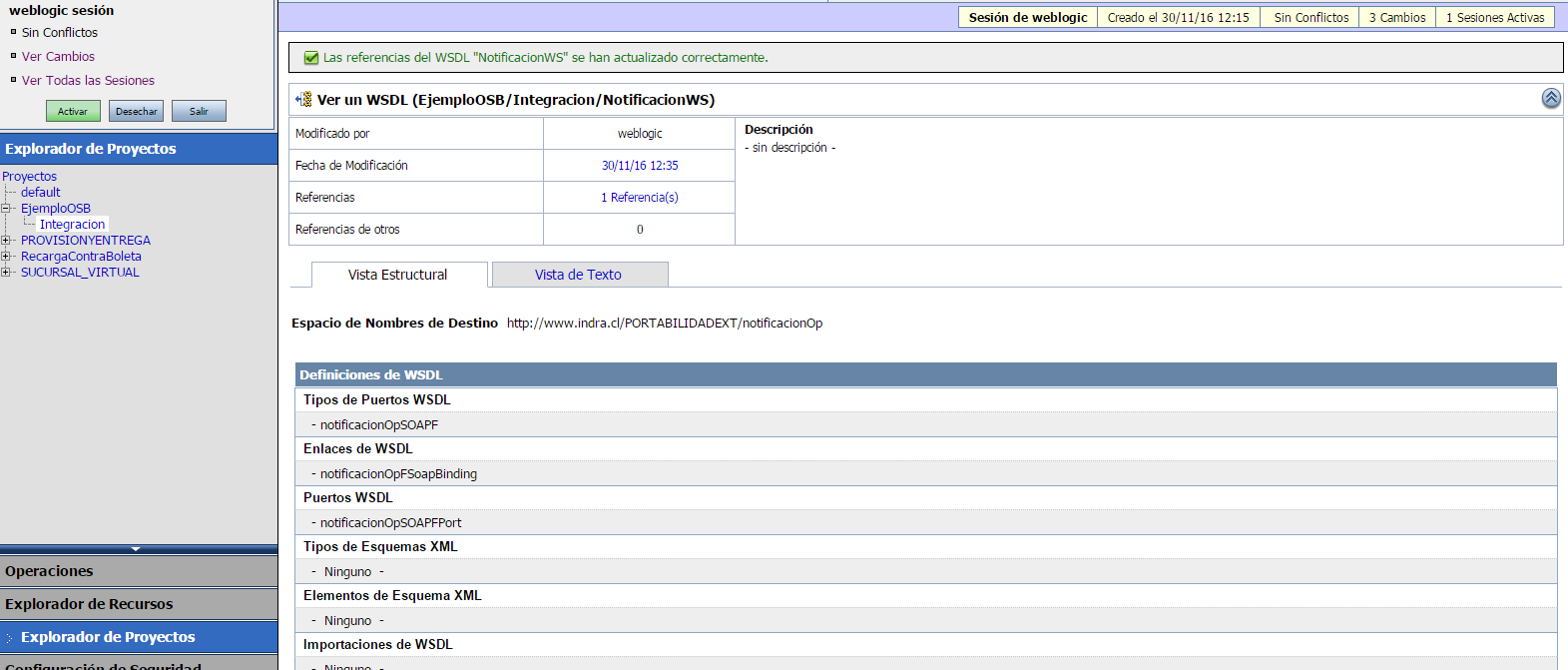
  
  
  
  


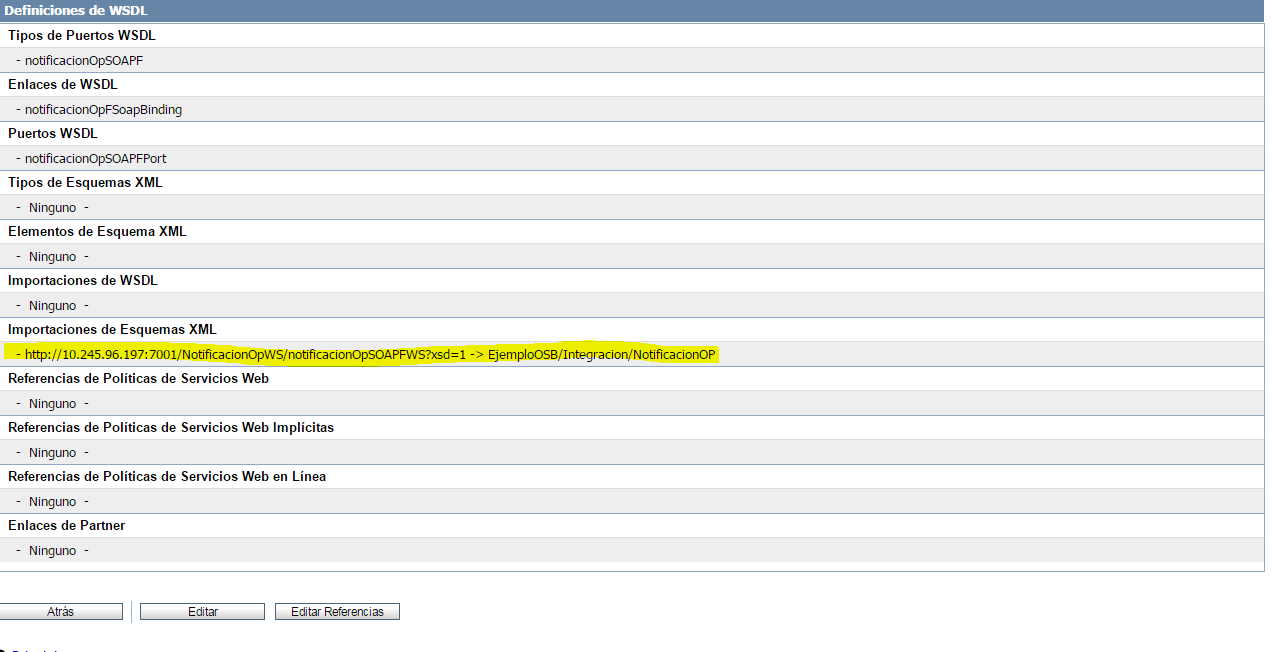


El conflicto se resuelve editando el enlace del XSD en el wsdl.  
  
Importante para ambientes de red intranet offline dejar las referencias locales y incluidas en el proyecto para que el servidor no pierda las referencias.



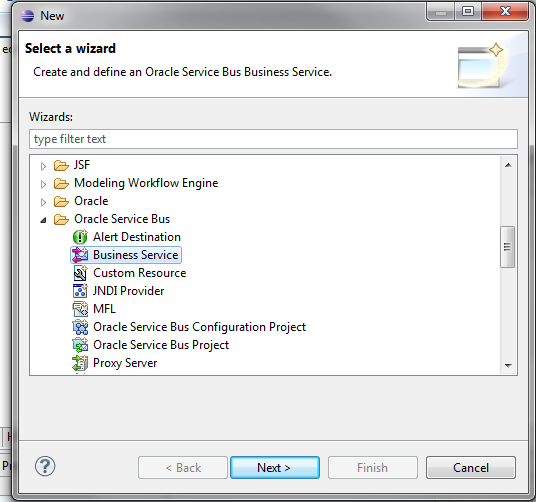


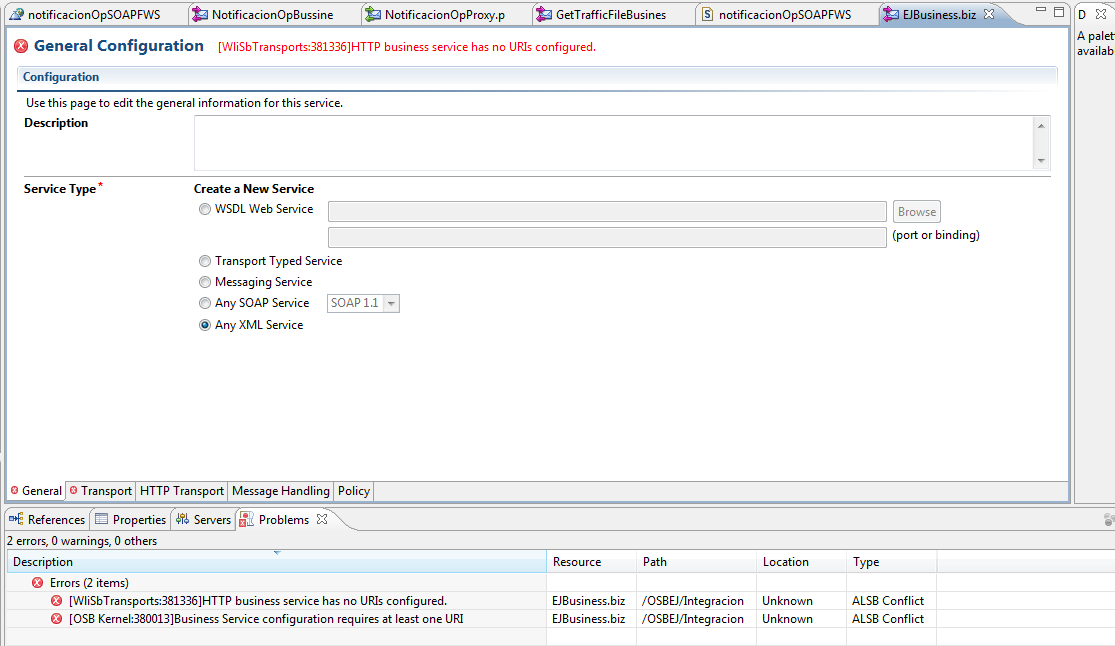
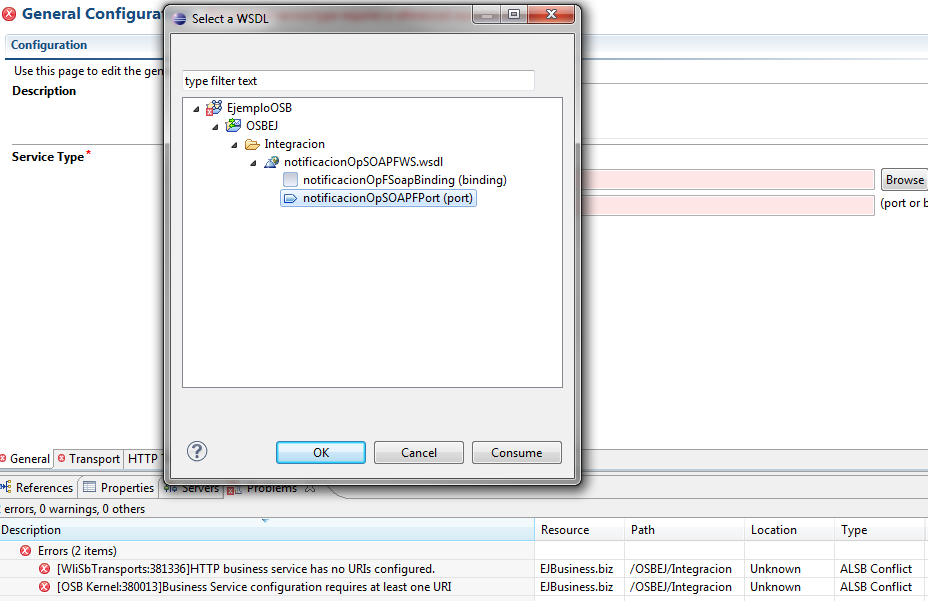
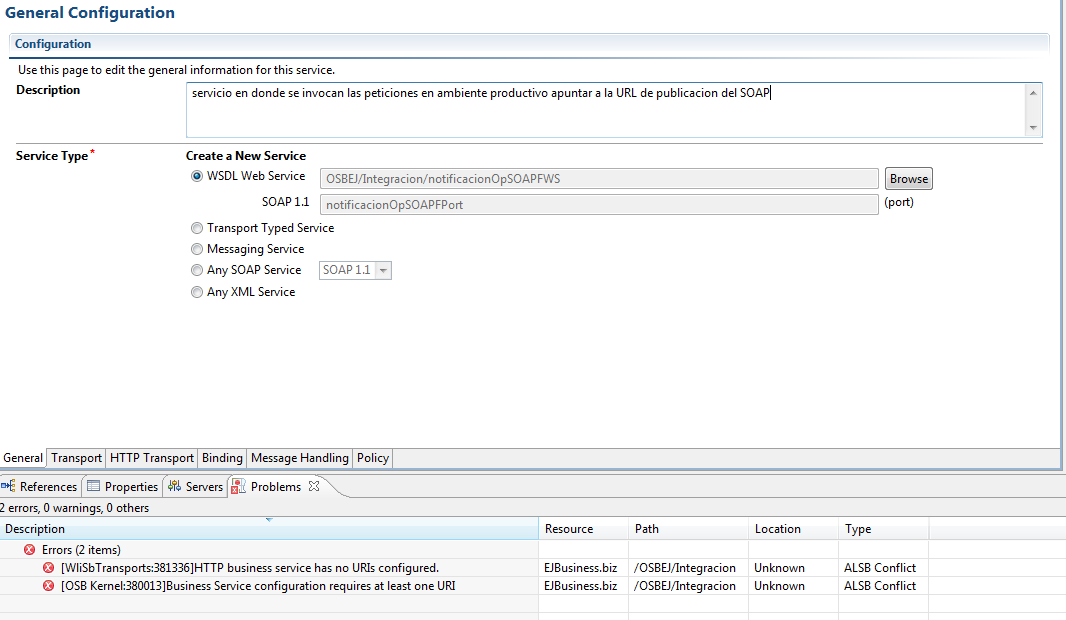


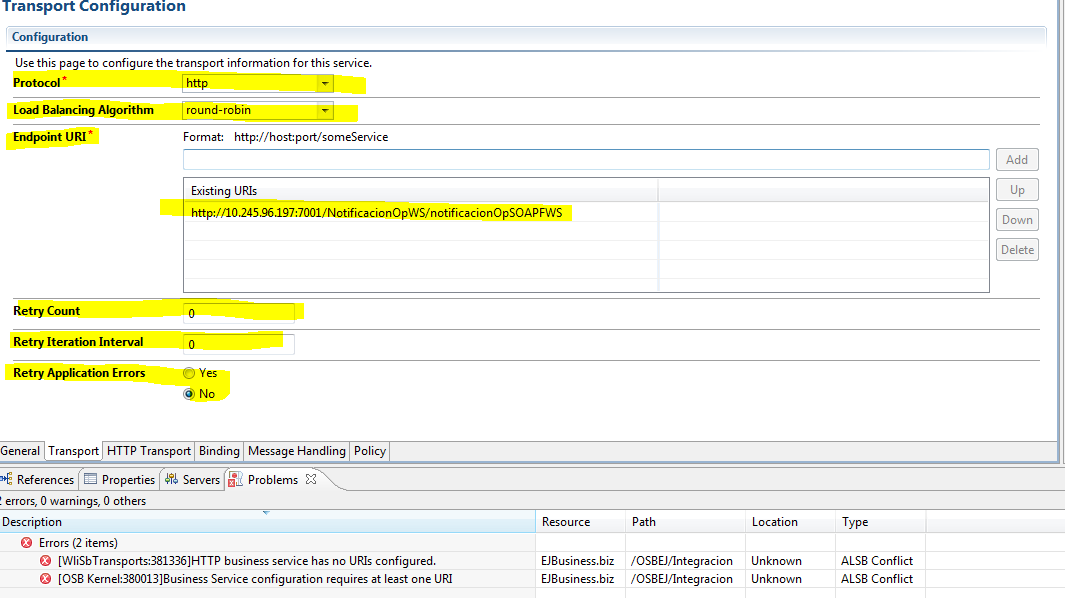
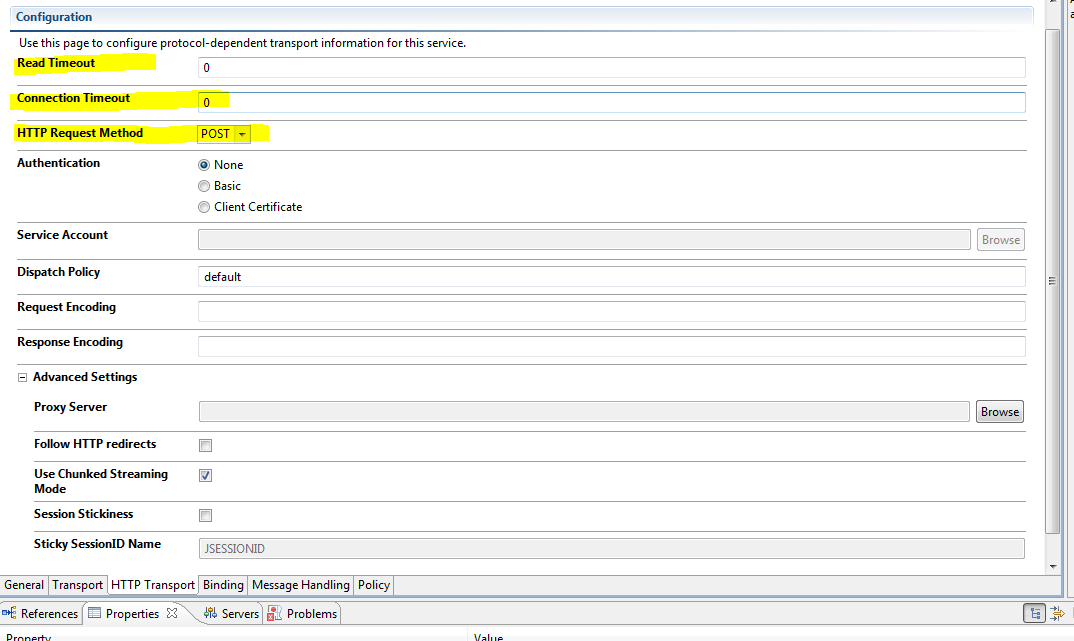


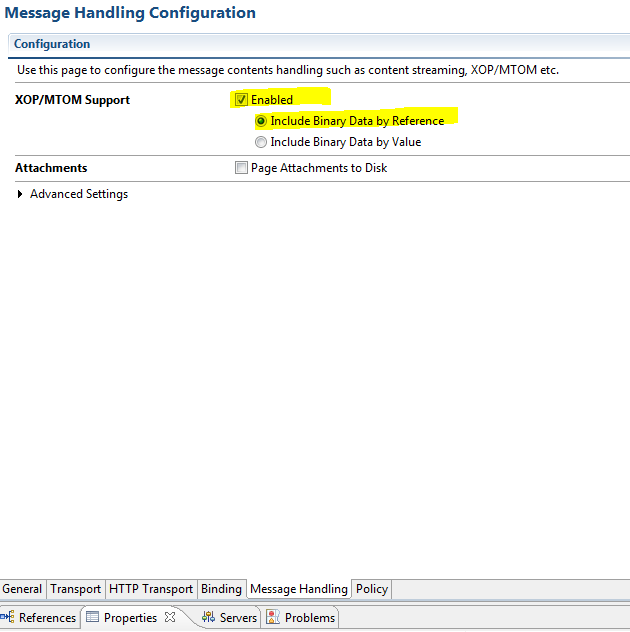


Se crea el Business el nombre debe seguir el standard del cliente.



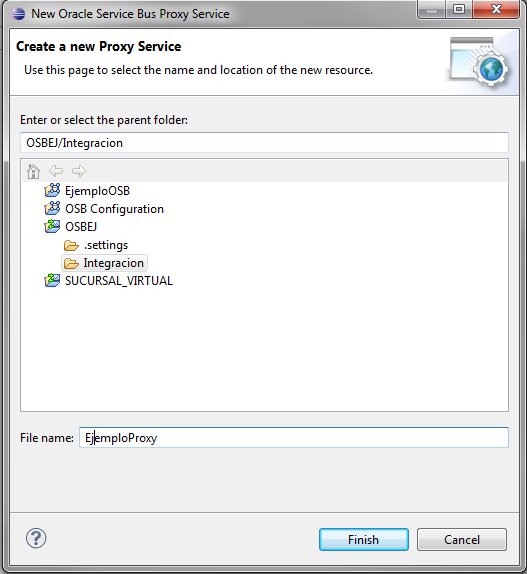
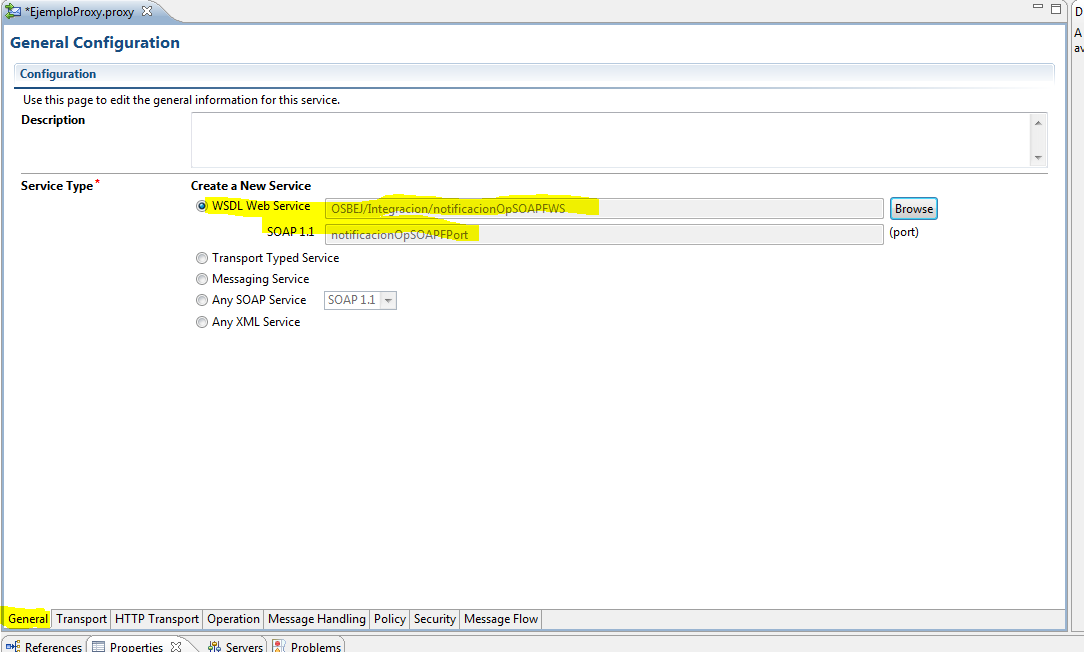
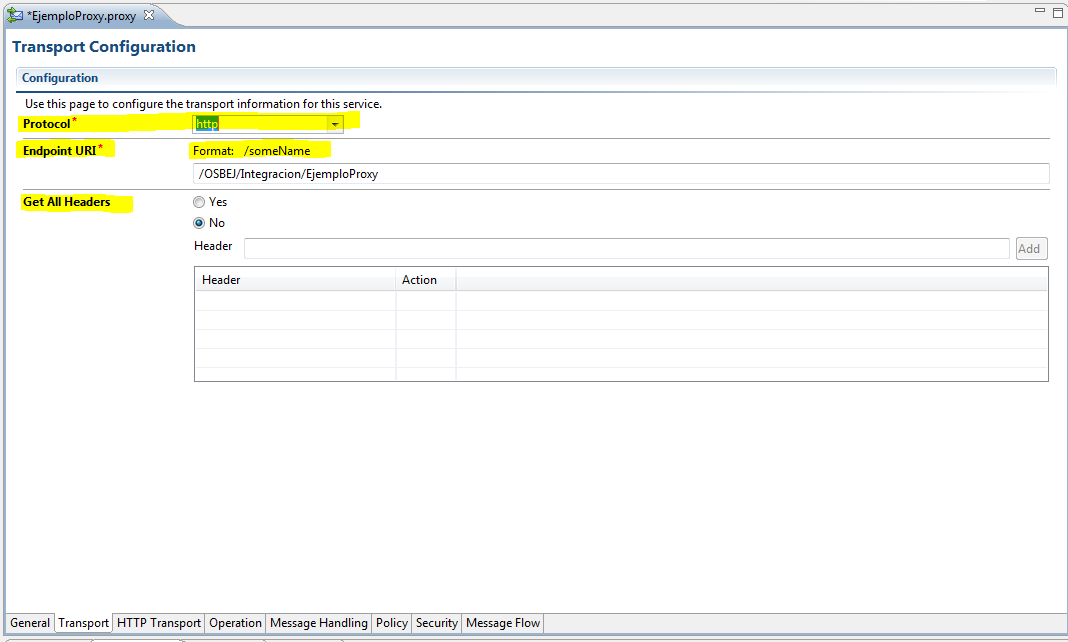
  
  
  
  
En esta pantalla se pueden configurar los reintentos y intervalos entre peticiones al EndPoint del SOAP, para servicios que no estén publicados en Weblogic y que no tengan Load Balancing se puede agregar uno por defecto.

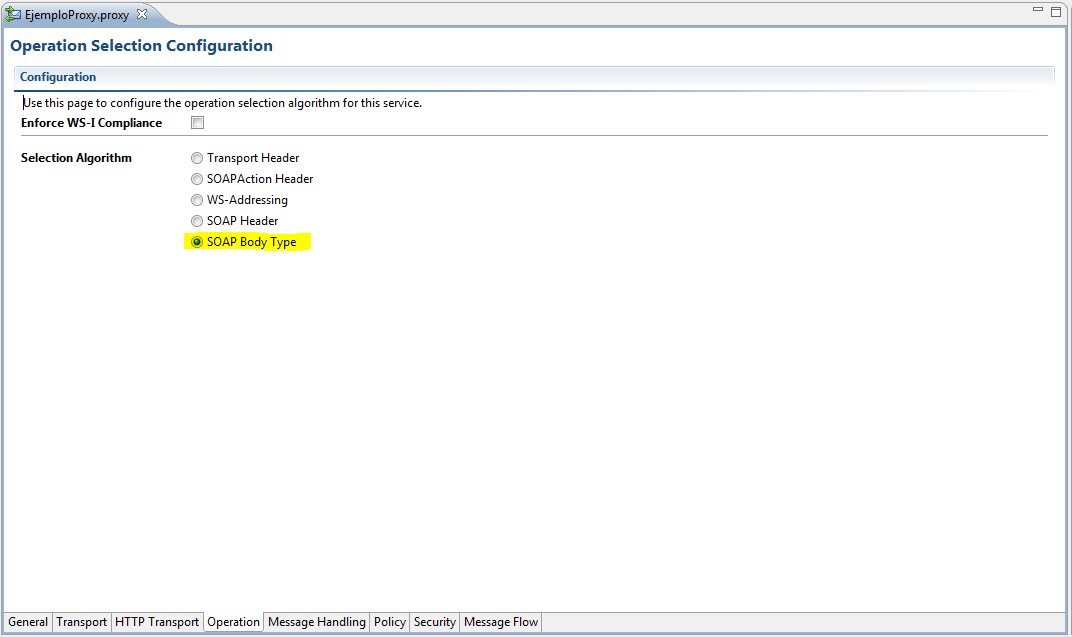
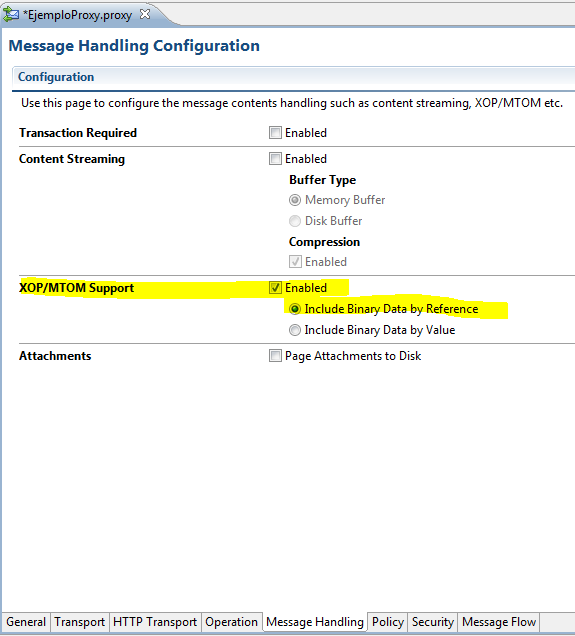
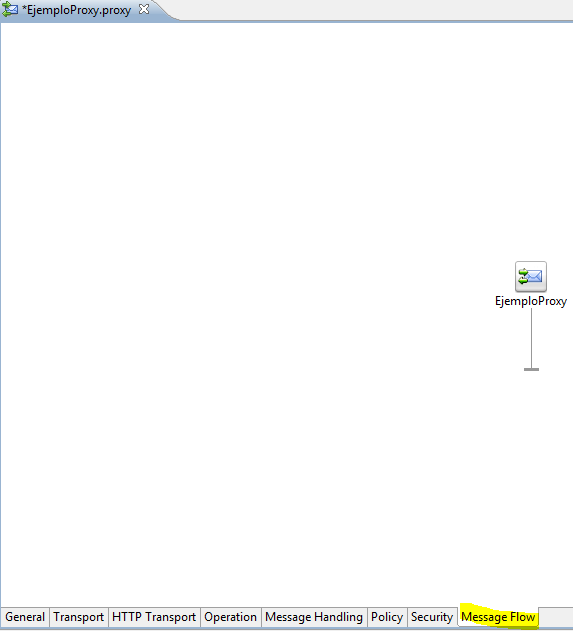
  


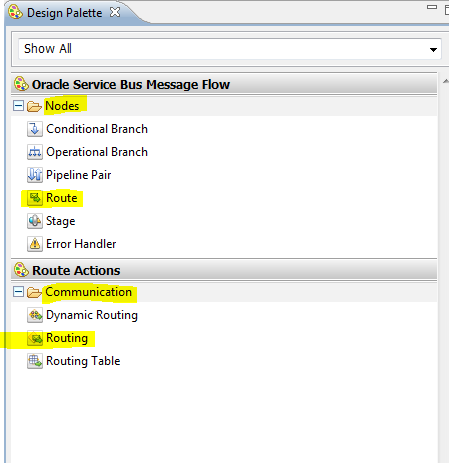
En la Pestaña de HTTP Transport se puede configurar el servicio en este ejemplo de integración SOAP a SOAP el Method es POST, ahora algo importante los valores Read Timeout es el valor en que el servidor espera por una respuesta a una petición y el tiempo de conexión son valores que siempre deben estar configurados, ya que permite dar estabilidad al servicio para no quedarse detenido esperando una respuesta a una petición. En caso de ser un Webservice con seguridad llenar pestaña de seguridad con el tipo de seguridad ej: Basic, Client Certificate.  
**Ahora en caso de que el WS SOAP trabaje con archivos debe llenarse la plantilla Message Handling**:  
   


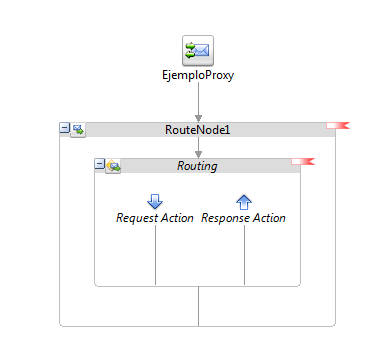
### Creación de Proxy:

En la capa de negocio se preparó la invocación al servicio, en esta capa proxy se puede estructurar una respuesta, manejo de errores, cambiar Response, manejar Cache a las peticiones, tratamiento de mensajes,etc.

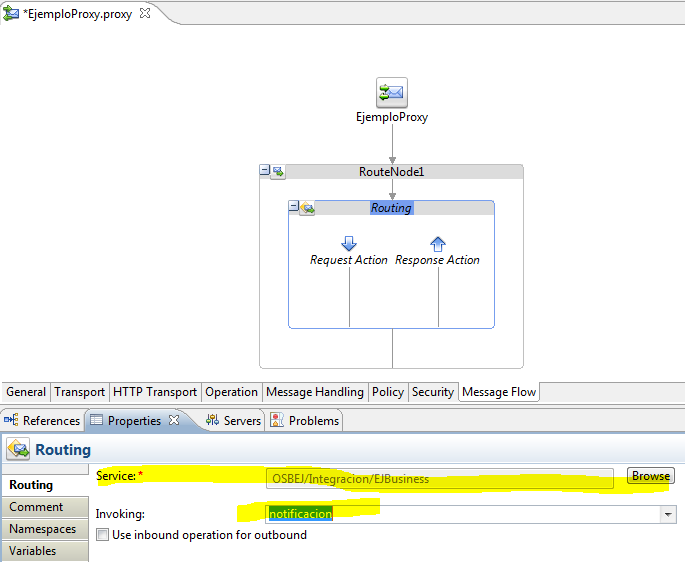
  
  
  
  
En esta penstaña de transport se configura la URL del punto final del servicio, es la URL expuesta del OSB:  
  
  
  
En esta pantalla se puede configurar el método de selección de la operación del WS, esto depende de la implementación del WS SOAP.

  
**Ahora si se el WS de origen consumido en el Business también tiene archivos, se debe activar en el proxy el soporte para XOP/MTOM**.  
  
  
Ahora después de configurar el tratamiento de mensajes se debe configurar el flujo de mensajes, para el caso de una integración entre SOAP a SOAP, el proxy inicial será:  
  
Seleccionar en el menú de paleta en el OEPE en la pestaña Nodes, el componente Route, luego en Communication el componente Routing en el flujo debe aparecer de esta forma:

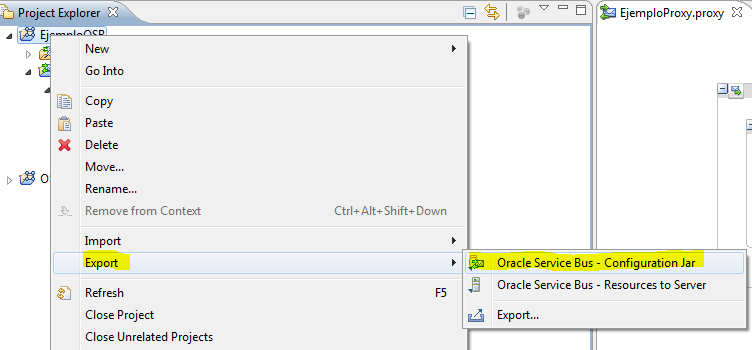


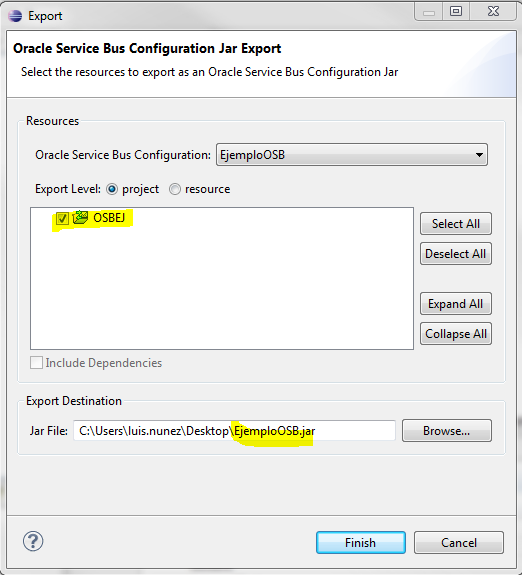


Ahora se debe editar el direccionamiento del Routing:

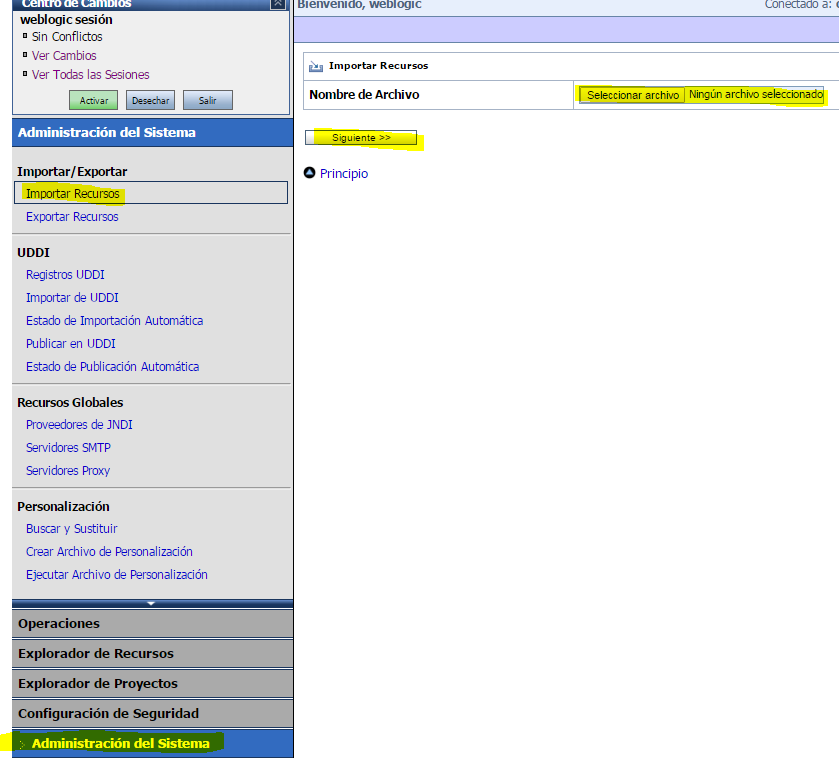


Ojo con la invocación en un webservice que tenga más de un método definido se debe invocar al método en específico, para que el proxy genere el Response correspondiente.  
  
  
Luego que la integración está completa y probada, se puede exportar a un Jar que se puede instalar en la consola de administración del OSB en la sección “Administración del Sistema”





### Instalación y prueba de servicio integrado al bus.



Ojo con las operaciones, en caso de un proyecto existente se debe tener presente con las sustituciones y borrado de componentes.

Las Uris de punto final del business solo se pueden configurar desde la consola de administración con la opción de remplazar variables.





Ejecutando la consola de pruebas de servicio para probar el direccionamiento del proxy:

