Documentación Técnica

Conector

Torre Ejecutiva Sur
Liniers 1324, piso 4
Montevideo – Uruguay
Tel/Fax: (+598) 2901.2929*
Email: contacto@agesic.gub.uy

www.agesic.gub.uy

Indice

1 Introducción

El Conector es una aplicación cliente generada y distribuida por AGESIC para ser instalado en los organismos que se conectan con la PGE, y que permite transformar un consumidor de web service plano (entrada) a los estándares de la plataforma PGE (WS-Security, WS-Trust, WS-Addressing).

El presente documento contiene la documentación técnica de la aplicación Conector, e incluye vistas de arquitectura en Modelo 4+1 (excluyendo vistas de procesos) y diseño de pipelines del ESB.

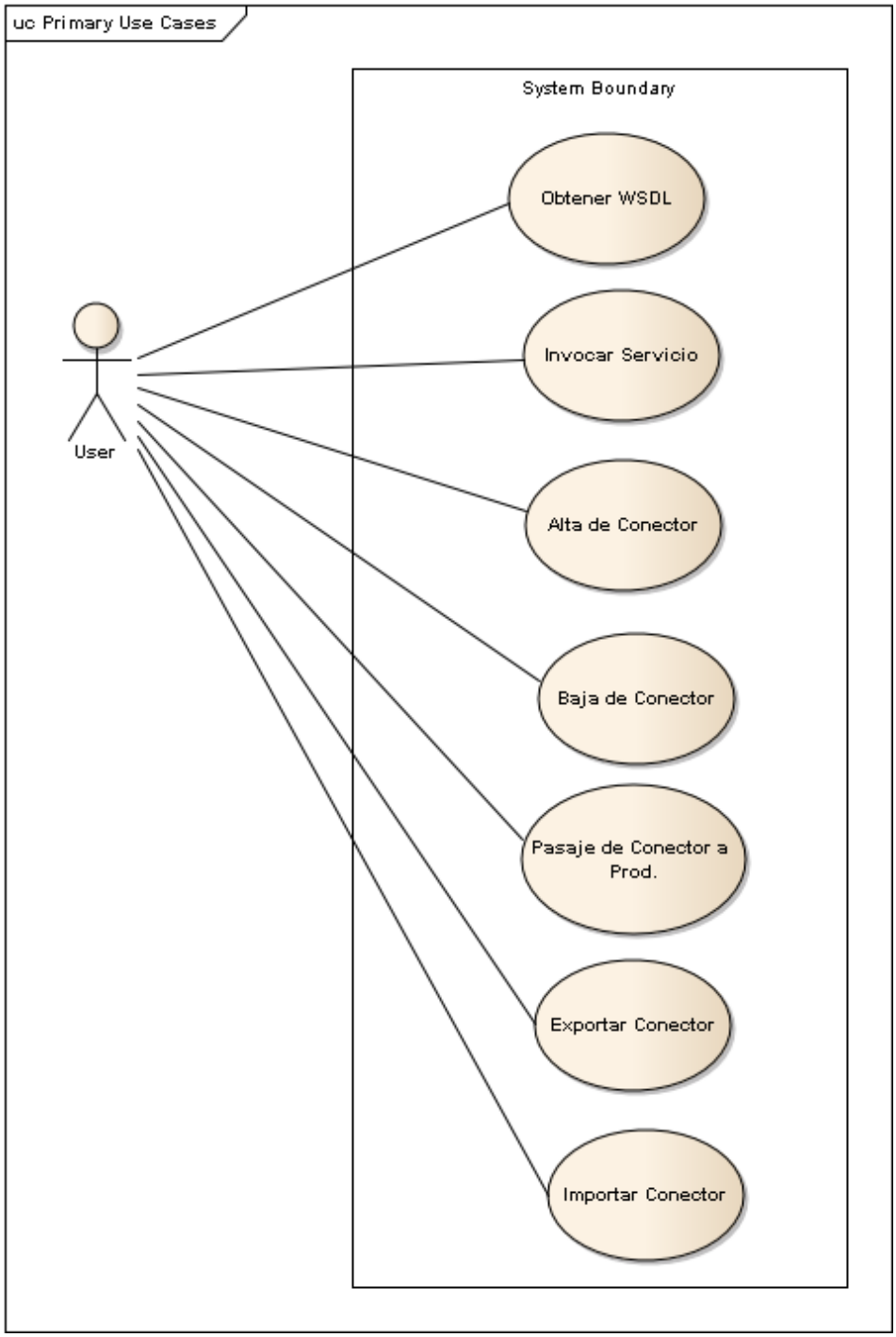
El documente se organiza de la siguiente forma:

- El segundo capitulo del documento define los casos de uso utilizados en el Modelo 4+1.
- El capitulo tres presenta el diagrama de clases, que corresponde a la vista lógica del modelo.
- El cuarto capítulo muestra el diagrama de componentes, que corresponde a la vista de implementación del modelo.
- El quinto capítulo contiene el diseño de pipelines del ESB (similar al diagrama de actividades asociado a la vista de procesos del modelo).
- Por último, el sexto capítulo contiene el diagrama de deploy del conector.

2 Casos de Uso

Los casos de uso que posee el sistema Conector, y que son utilizados como núcleo del modelo 4+1, se enumeran en el diagrama de casos de uso, mostrado en la siguiente sección. Más adelante, se describe cada uno de ellos, y se presenta su diagrama de interacción.

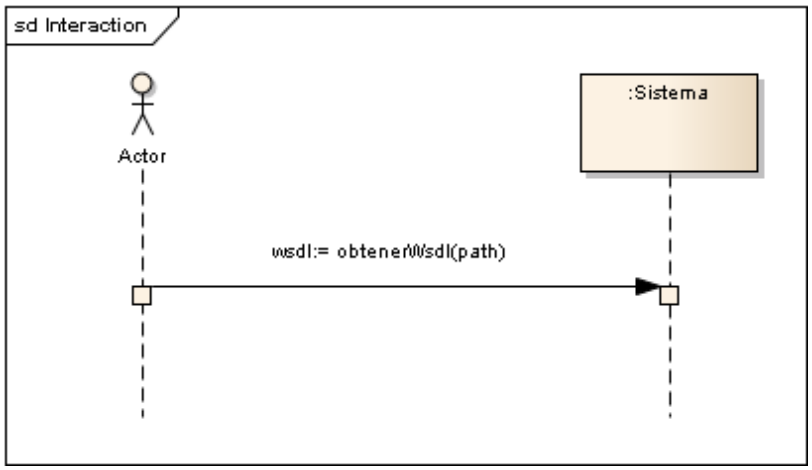
2.1 Diagramas de Caso de Uso



2.1.1 Obtener WSDL

Caso de uso: Obtener WSDL
Actor: Usuario del Conector
Curso Normal:
1) El usuario realiza un pedido al conector a través de un http post, indicando como path el definido en el conector deseado, y finalizando la uri con ?wsdl
2) El sistema busca el conector cuyo path definido sea igual al recibido.
3) El sistema genera una traza de log, de manera de registrar el pedido. A su vez, obtiene el wsdl del conector encontrado.
4) El campo de soapaddress del wsdl es modificado para que la url apunte al conector, y el wsdl es devuelto al usuario como respuesta del http post.

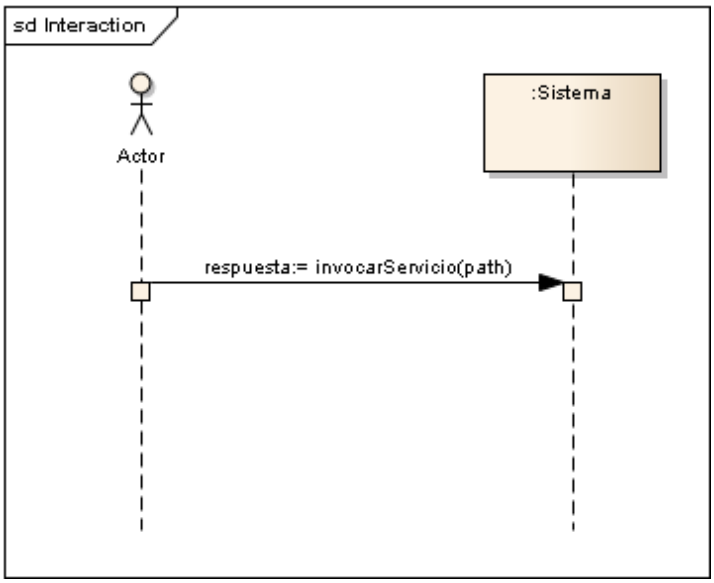
Diagrama de interacción:



2.1.2 Invocar Servicio

Caso de uso: Invocar Servicio
Actor: Usuario del Conector
Curso Normal:
1) A partir del wsdl obtenido en el Caso de Uso **Obtener WSDL,** el usuario realiza una invocación al conector a través de un http post, utilizando como endpoint el especificado por el wsdl.
2) El sistema valida que exista el conector, y obtiene los datos del mismo.
3) El sistema genera una traza de log, de manera de registrar el pedido. A su vez, agrega al mensaje enviado por el cliente los campos del estándar WS-Addressing, necesarios para interactuar con la PGE.
4) El sistema obtiene del servidor STS de Agesic un token SAML, y agrega los campos de WS-Security al mensaje del cliente, de manera de poder interactuar con la PGE.
5) El sistema realiza la invocación al servicio final, y obtiene la respuesta.
6) El sistema envía una traza, de manera de registrar la llegada de la respuesta. A su vez, el sistema devuelve al cliente el resultado obtenido.

Diagrama de interacción:

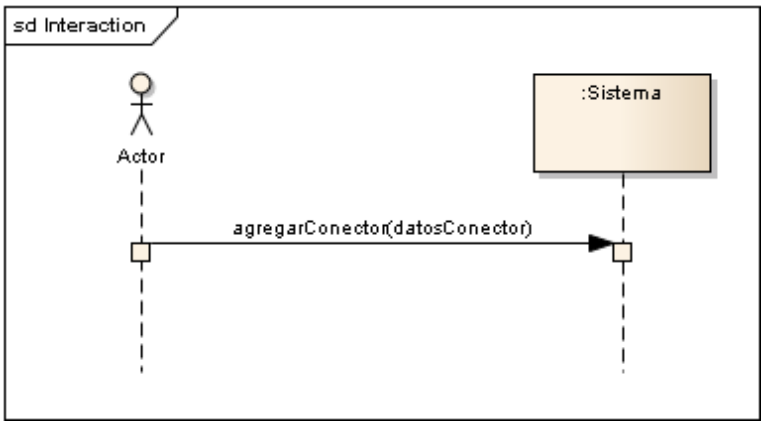


2.1.3 Alta de Conector

Caso de uso: Alta de Conector
Actor: Usuario del Conector
Curso Normal:
1) El usuario ingresa al sistema, y se selecciona el ambiente donde desea crear el conector (Test, Producción). Luego, utiliza el botón de Nuevo Conector.
2) El sistema muestra una pantalla con el formulario de creación de un conector. Pide al usuario completar los siguientes datos obligatorios: nombre, tipo, descripción, path, wsaTo, username, organismo, tipo de token y wsdl. También permite seleccionar el uso de cache de tokens y configuración local. En este último caso, se debe ingresar todos los datos de keystores y truststores.
3) El usuario ingresa todos los datos requeridos por el sistema, y utiliza el botón de guardar.
4) El sistema valida los datos ingresados, y en caso de haber completado todos los datos

obligatorios, procede a guardar el conector. Además, el sistema cargará las operaciones encontradas en el wsdl en una tabla indicando wsaAction y nombre de la operación leída. En caso de haber detectado errores de validación de keystores, truststores, o en el parseo del wsdl, se mostrará un mensaje de advertencia en la pantalla de listado de conectores.

Diagrama de Interacción:

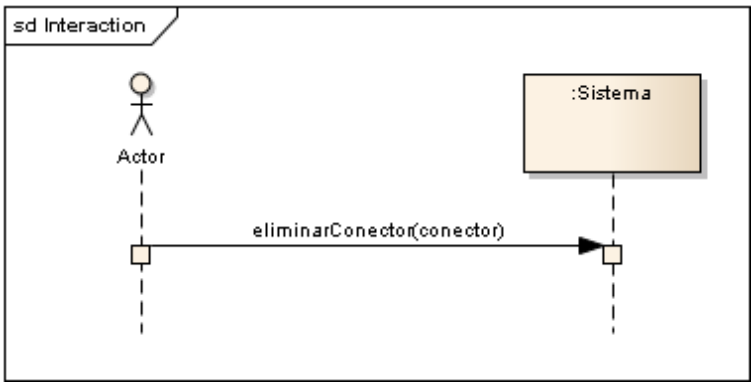


2.1.4 Baja de Conector

Caso de uso: Baja de Conector
Actor: Usuario del Conector
Curso Normal:
1) El usuario ingresa al sistema, y se selecciona el ambiente (Test, Producción) donde se encuentra el conector a eliminar.
2) El usuario selecciona el conector a eliminar, utilizando el botón de Eliminar Conector
3) El sistema pide confirmación de la eliminación del conector.

4) El usuario confirma la acción.
5) El sistema procede a borrar el conector, y elimina cualquier dependencia con su correspondiente conector de producción (en caso que aplique).

Diagrama de Interacción:

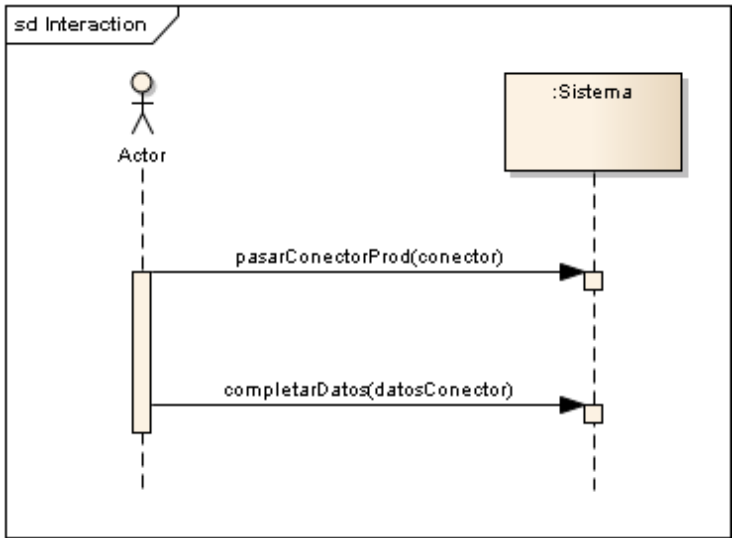


2.1.5 Pasaje de Conector a Producción

Caso de uso: Pasaje de Conector a Producción
Actor: Usuario del Conector
Curso Normal:
1) El usuario ingresa al sistema y selecciona el ambiente Test.
2) El sistema lista en pantalla todos los conectores de Test.
3) El usuario elige el conector y hace click en el boton de Editar que esta en la misma fila del conector.
4) El sistema muestra en pantalla el formulario de edición con los datos del conector.

5) El usuario hace click en el botón Pasar a Prod.
6) El sistema muestra en pantalla el formulario de creacion de un nuevo conector con los datos del conector de test con la excepción del ambiente que se carga por defecto el de Producción.
7) El usuario puede modificar los datos que desee y luego hace click en Salvar.
8) El sistema muestra en pantalla los datos del nuevo conector.

Diagrama de interacción:

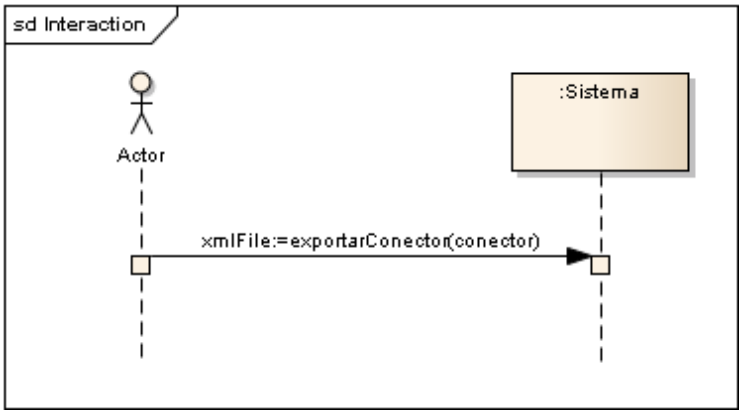


2.1.6 Exportar Conector

Caso de uso: Exportar Conector
Actor: Usuario del Conector
Curso Normal:

1) El usuario ingresa al sistema, y se selecciona el ambiente del conector a exportar (Test, Producción).
2) El usuario elige el conector a exportar de la lista y da click en el enlace “Exportar” de dicho conector.
3) El sistema muestra en pantalla el popup de descarga para guardar el archivo xml del conector. Este archivo contiene toda la información necesaria del conector para que luego pueda ser importado.

Diagrama de interacción:

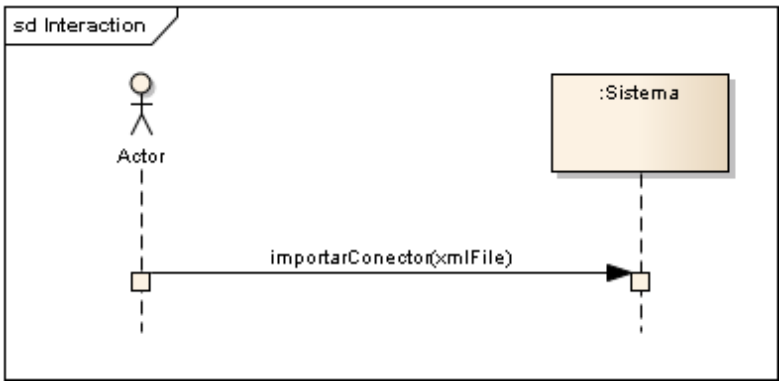


2.1.7 Importar Conector

Caso de uso: Importar Conector
Actor: Usuario del Conector

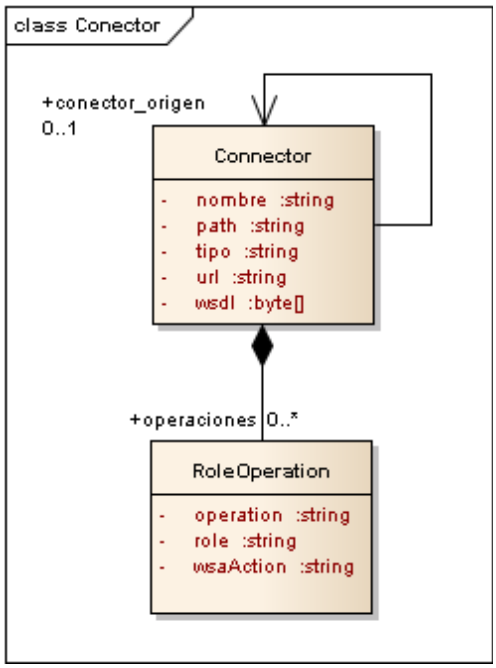
Curso Normal:
1) El usuario ingresa al sistema y hece click en el selector de Archivos que se encuentra a la derecha del boton Importar.
2) El sistema muestra en pantalla el explorador de archivos para elegir el archivo xml con la información del conector.
3) El usuario elige el archivo xml del conector y hace click en el boton Importar.
4) El sistema muestra en pantalla en conector importado.

Diagrama de interacción:



3 Diagrama de Clases

Las principales entidades que componen el sistema Conector se pueden apreciar en el siguiente diagrama de clases:

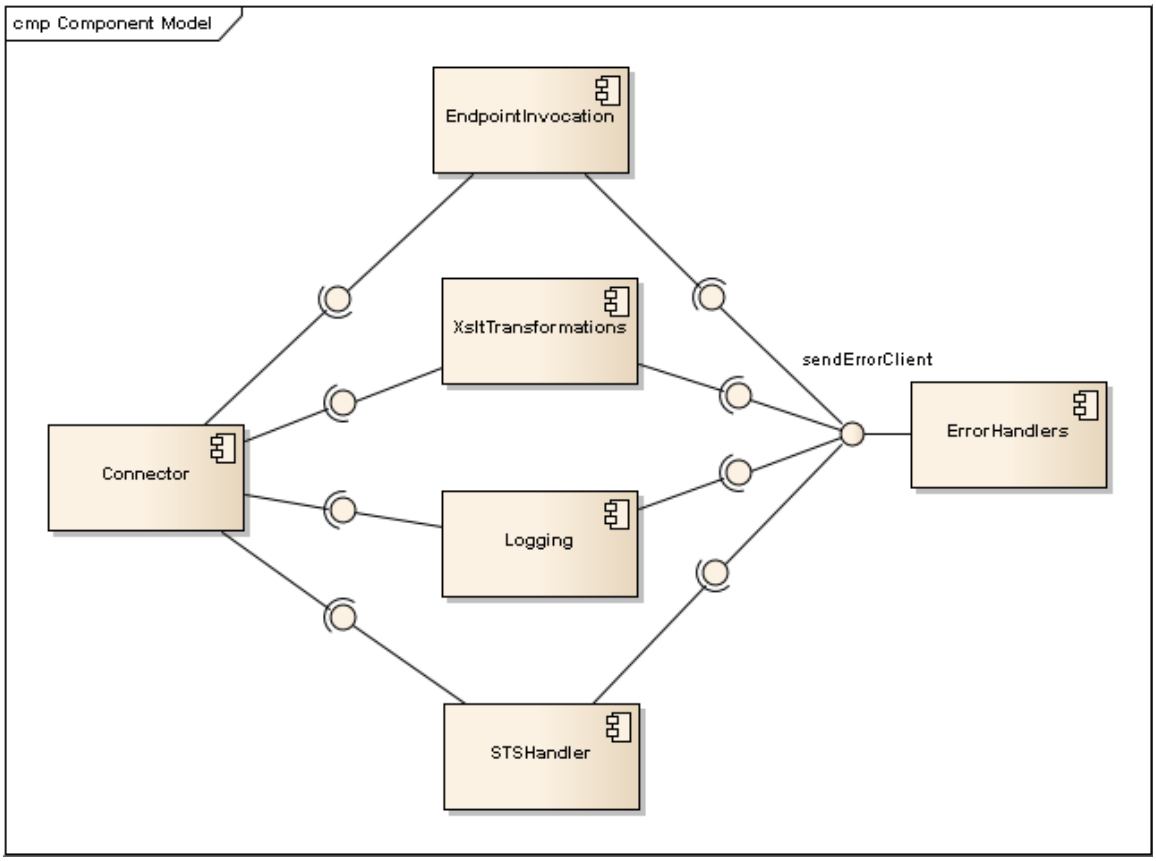


El sistema cuenta con dos clases principales; *Connector* contiene la información de los conectores que mantiene el sistema. Además de los campos que muestra el diagrama, la clase posee otros, tales como datos de configuración (utilizar cache tokens, configuración global, etc.) y certificados. La clase *RoleOperation* contiene la información de las operaciones disponibles para cada conector. La información se obtiene al parsear el wsdl que se adjunta a un conector.

Se puede observar que un *Connector* posee una referencia a si mismo, y esto se debe a que un *Connector* de tipo Producción puede tener referenciado el *Connector* de Test que lo originó.

4 Diagrama de Componentes

Los componentes que forman parte del sistema conector, que implementan los casos de uso planteados en el capítulo dos, se pueden observar en el siguiente diagrama de componentes.



A continuación se describe brevemente cada uno de los componentes:

- Connector: Contiene la lógica asociada al manejo de conectores. Por ejemplo, obtener la información de un conector, realizar el parseo de archivos wsdl, etc.

- EndPointInvocation: Este componente es el encargado de realizar las invocaciones a los servicios finales, a través de los protocolos HTTP y HTTPS.

- XsltTransformations: Permite realizar transformaciones XSLT. Estas son usadas por ejemplo, para agregar los *headers* de *WS-Addressing*.

- Logging: Componente encargado de generación y envío de trazas, así como también realizar tareas de logging.

- STSHandler: Permite obtener el token SAML (mediante la conexión con un servidor STS) de manera de autenticarse con la PGE.

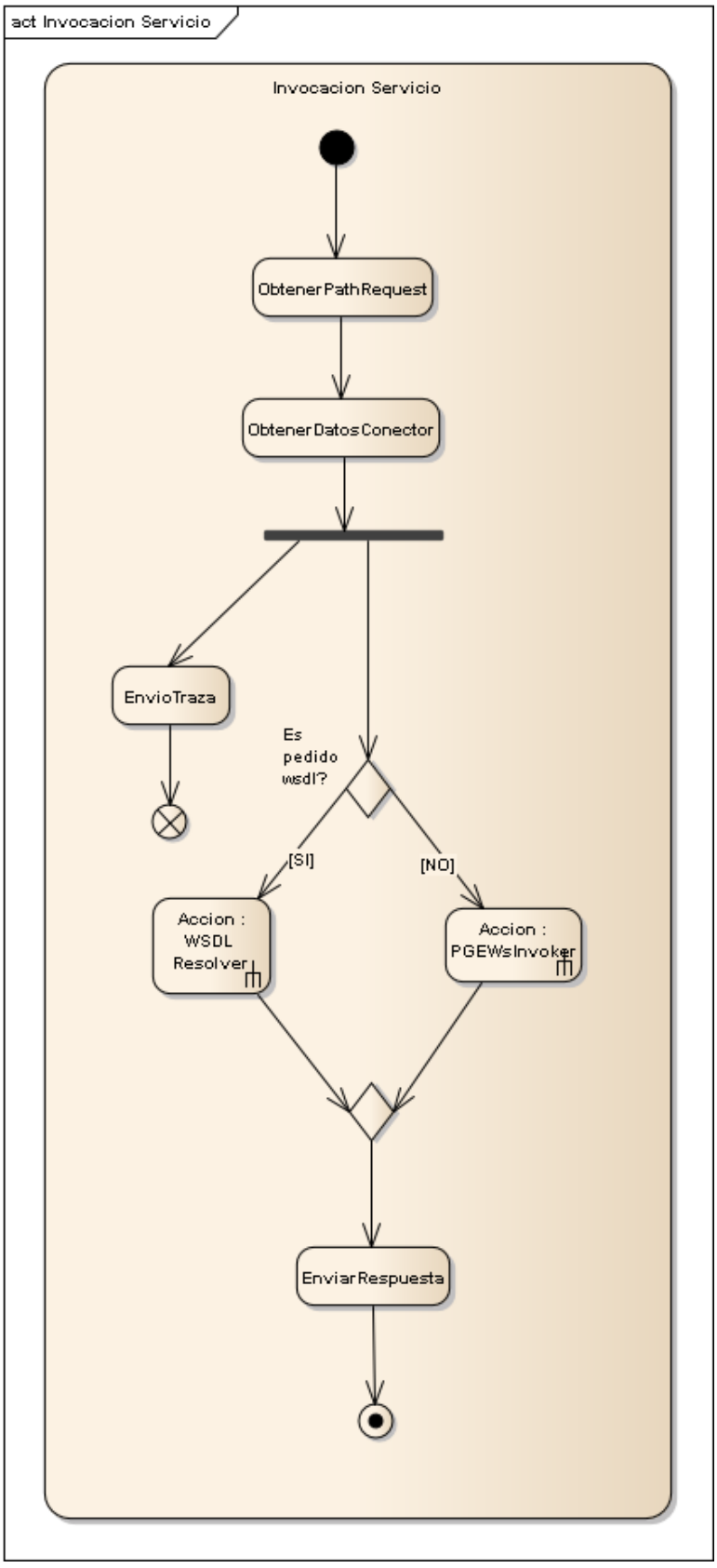
- ErrorHandlers: En caso de encontrar errores en la invocación de un servicio, este componente se encarga de notificar al cliente a través de un mensaje *Soap Fault*.

5 Diseño de Pipelines del ESB

En esta sección se presenta el diseño de pipelines del esb, mediante el uso de diagramas de actividad. Estos diagramas muestran la ejecución de los casos de uso de Obtener WSDL e Invocación de Servicio, desde el punto de vista del pipeline ESB del Conector. Para cada diagrama presentado, se describe brevemente las actividades que lo componen.

A continuación se presenta el diagrama que corresponde a la recepción de un mensaje, el cual puede ser un pedido de obtención de wsdl, o la invocación de un servicio.

Diagrama de actividades:

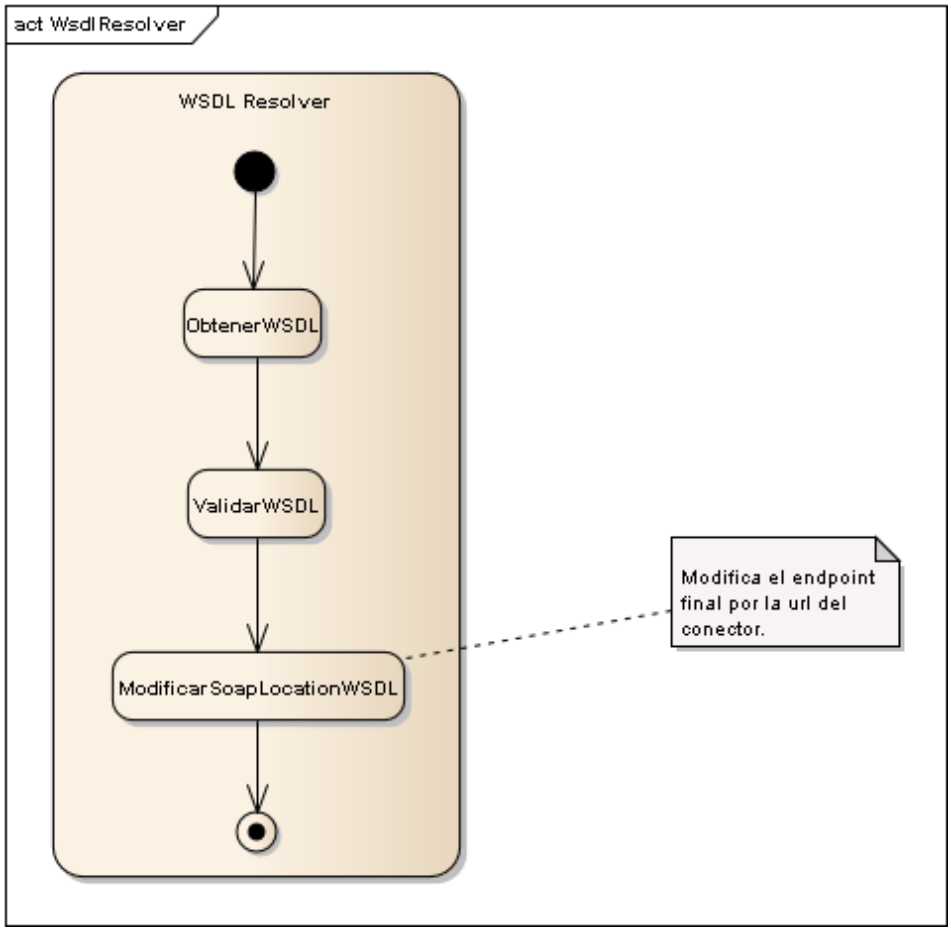

act Invocacion Servicio
Invocacion Servicio
ObtenerPathRequest
ObtenerDatosConector
EnvioTraza
Es pedido wsdl?
[SI]
[NO]
Accion : WSDL Resolver
Accion : PGEWsInvoker
EnviarRespuesta

Se puede observar que las primeras acciones obtienen el path del request, y con dicho path se obtiene el conector asociado. Luego de obtener los datos, se envía una traza de manera de notificar el arribo de una petición, y concurrentemente se determina el tipo de operación a realizar. En caso de detectar un *?wsdl* en el request, el sistema devuelve el wsdl del conector, ejecutando la acción Action: WSDL Resolver. En caso contrario, el pedido recibido corresponde a una invocación de servicio del conector, y por lo tanto se ejecuta la acción Action: PGEWsInvoker. Más adelante se detallan los diagramas de actividad de ambas funcionalidades.

Luego de ejecutada la operación deseada (obtener wsdl ó invocación de servicio), se observa que se envía la respuesta generada por la acción al cliente, y se finaliza el proceso.

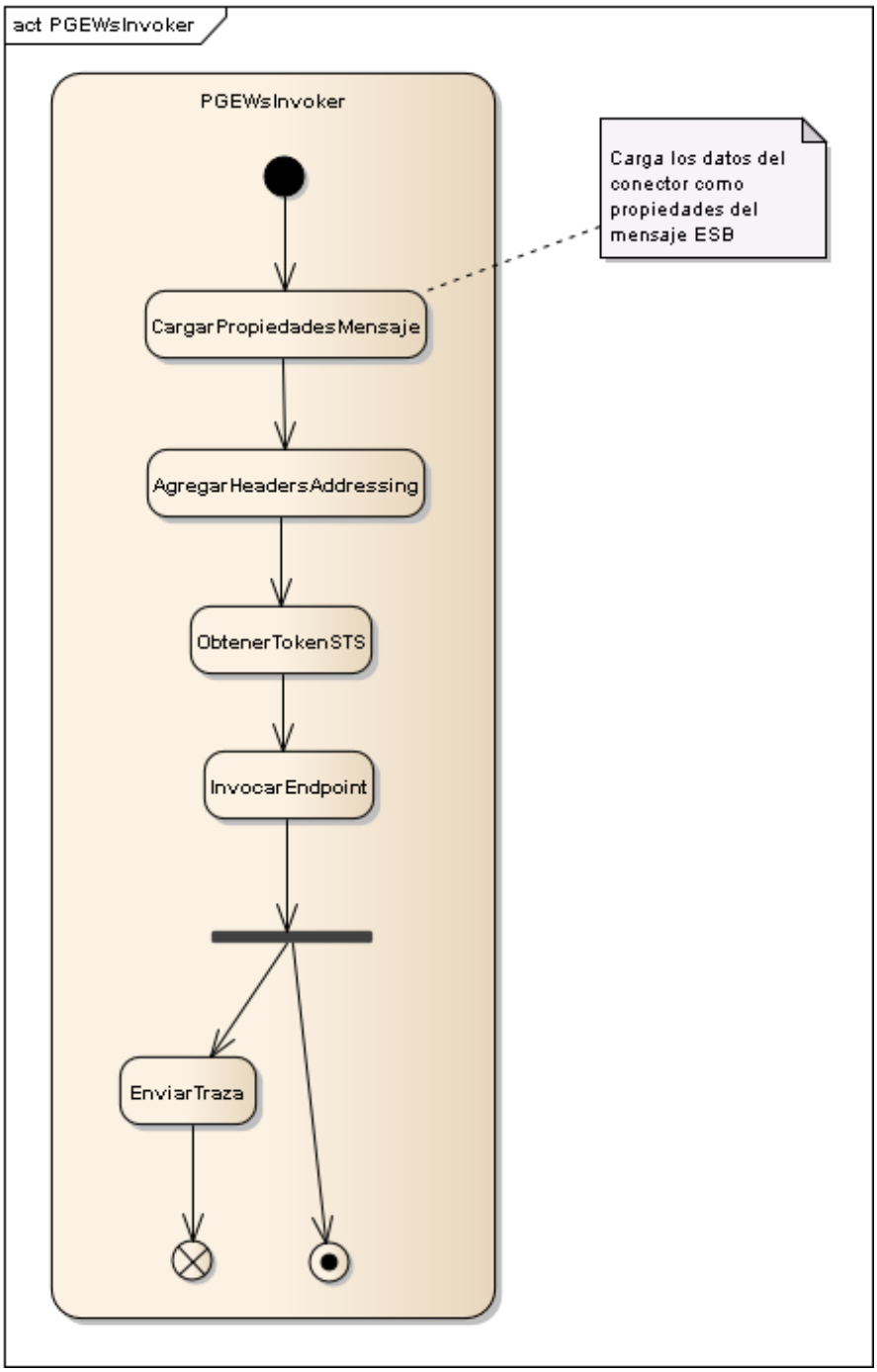
A continuación se presentan los diagramas de las acciones Action: WSDL Resolver y Action: PGEWsInvoker mencionados anteriormente.

Action: WSDL Resolver:



Este diagrama se encuentra compuesto por 3 acciones. *ObtenerWSDL* obtiene el contenido del archivo wsdl asociado al conector. Luego, *ValidarWSDL* verifica que se haya obtenido correctamente el contenido del wsdl, y por último *ModificarSoapLocationWSDL* modifica el endpoint del servicio especificado en el wsdl, de manera que apunte al conector. A modo de ejemplo, si el path del conector fuera *conectorEjemplo*, y el conector corresponde a uno de testeo, entonces la url del endpoint sería: http://servidorConector:9800/conectorEjemplo donde *servidorConector* es el host donde se encuentra deployado la aplicación Conector, y 9800 corresponde al puerto de testeo por defecto.

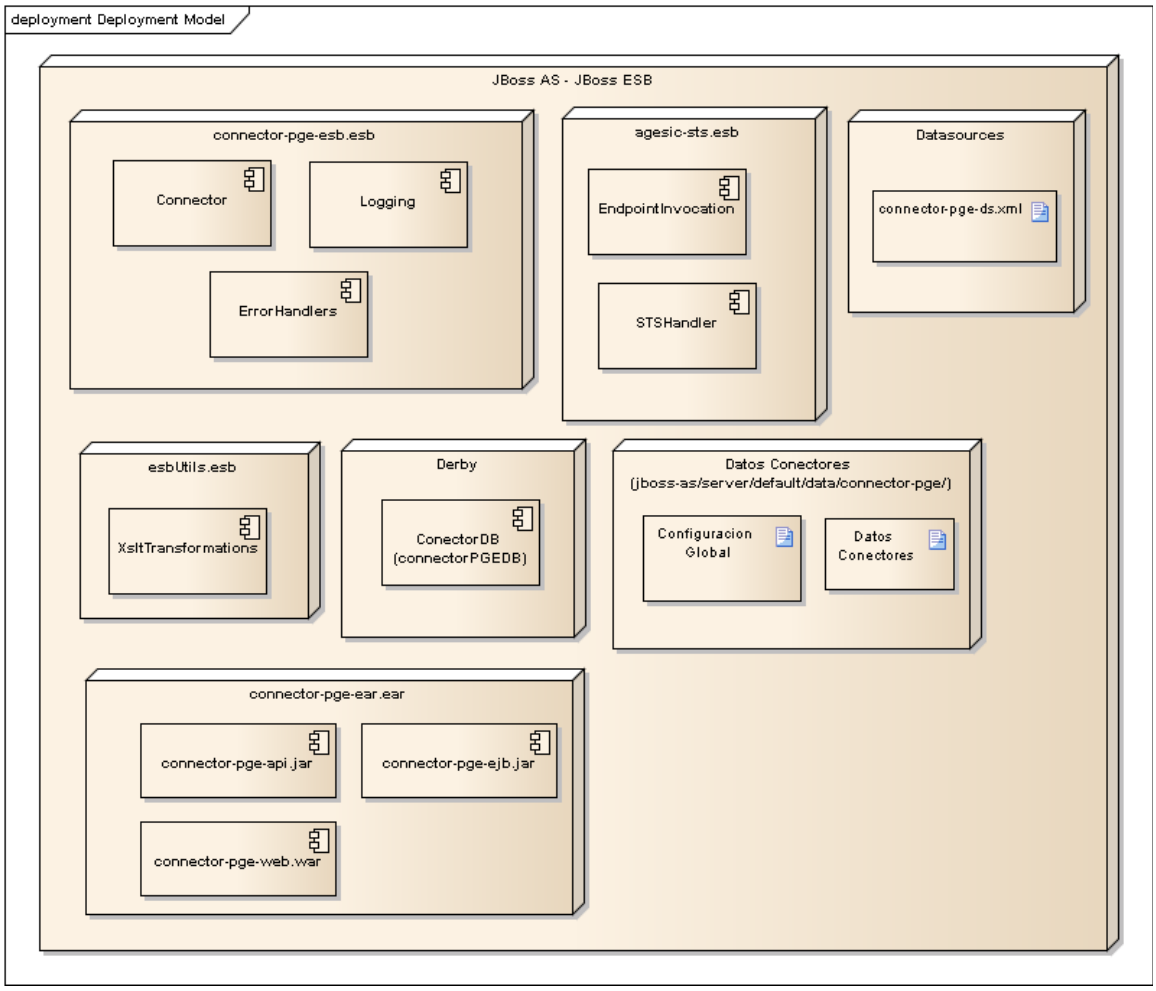
Action: PGEWsInvoker



En el diagrama, se observa que la primera acción *CargarPropiedadesMensaje* carga los datos del conector en las propiedades del mensaje ESB, por ejemplo wsaTo, nombre del conector, información de keystores, etc. Estos datos son utilizados más adelante por el resto de las acciones que componen el proceso. La segunda acción *AgregarHeadersAddressing* realiza transformaciones XSLT de manera de agregar la información requerida por *WS-Addressing,* tales como *wsa:Action* y *wsa:To*. Luego, *ObtenerTokenSTS* se encarga de obtener un token SAML para *WS-Security*, realizando una invocación al servidor STS de la PGE. Una vez terminado el mensaje para ser enviado a la plataforma, se utiliza la acción *InvocarEndpoint*, la cual se encarga de invocar el servicio especificado por el conector. Una vez recibida la respuesta, se envía una traza de manera de notificar este evento, y de manera concurrente se devuelve el resultado al cliente.

6 Diagrama de Deploy

A continuación se presenta el diagrama de deploy del Conector.



Los componentes que se observan en el modelo de deploy son los siguientes:

- connector-pge-esb.esb: Aplicación esb del conector. Contiene las acciones del conector descritas en el documento, así como también los módulos de *Logging* y *ErrorHandlers*.

- agesic-sts.esb: Aplicación esb que contiene la lógica asociada a invocación del endpoint, y manejo de tokens SAML.

- esbUtils.esb: Aplicación esb que contiene acciones utilitarias para el conector. Dentro de ellas, se encuentra el manejo de transformaciones XSLT.

- Derby: El conector utiliza Apache Derby como RDBMS.

- Datasources: Dentro de este componente, se encuentra el datasource del conector *connector-pge-ds.xml*.

- Datos Conectores: Este componente muestra las carpetas donde se guarda la información de los conectores, así como también los certificados para la configuración global.

- connector-pge-ear.ear: Aplicación Java EE que permite realizar las tareas administrativas del sistema Conector. Se encuentra compuesto por *connector-pge-web.war* (capa de presentación), *connector-pge-ejb.jar* (capa de lógica de negocios) y *connector-pge-api.jar* que contiene definiciones de interfaces y clases globales utilizadas por otros módulos, entre ellos connector-pge-esb.esb.