Conector PGE

Manual de configuración y administración

Área de Tecnología

División Arquitectura

**Historial de Revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Cambios | Autor |
| 16/01/2018 | 1.0 | Creación del documento | Equipo Pyxis |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Tabla de contenido**

[**1** **Introducción** 3](#_Toc503828482)

[**2** **Instalación y configuración** 5](#_Toc503828483)

[2.1 **Configuración básica** 5](#_Toc503828484)

[**2.1.1** **Configuración básica componente web** 5](#_Toc503828485)

[**2.1.2** **Configuración básica componente Integration** 6](#_Toc503828486)

[2.2 **Ejecución** 7](#_Toc503828487)

[**3** **Administración de conectores** 11](#_Toc503828488)

[3.1 **Acceso** 11](#_Toc503828490)

[3.2 **Crear un nuevo Conector** 13](#_Toc503828491)

[3.3 **Ver Conector** 24](#_Toc503828492)

[3.4 **Editar un conector existente** 25](#_Toc503828493)

[3.5 **Eliminar un conector existente** 28](#_Toc503828494)

[3.6 **Ver WSDL** 29](#_Toc503828495)

[3.7 **Exportación de un conector** 31](#_Toc503828496)

[3.8 **Importación de Conector** 33](#_Toc503828497)

[3.9 **Remplazo de keystores** 34](#_Toc503828498)

[3.10 **Pasaje a producción** 35](#_Toc503828499)

[3.11 **Configuración global** 36](#_Toc503828500)

[**4** **Respaldo y recuperación** 38](#_Toc503828501)

[**5** **Anexo I – Creación de archivo Zip** 39](#_Toc503828502)

[**6** **Anexos II – Librerías Java** 40](#_Toc503828503)

[**7** **Anexo III – Puertos del conector** 41](#_Toc503828504)

# **Introducción**

Cuando un organismo desea consumir un servicio web expuesto en la Plataforma de Gobierno electrónico (PGE), hay varios aspectos que deben ser considerados. Entre ellos, se destacan el marco legal y técnico, intercambio seguro de información, y la participación en una arquitectura orientada a servicios (SOA). Estos aspectos requieren de cierto grado de madurez tecnológica, que muchas veces no se tiene en los organismos. A continuación, se listan las tecnologías necesarias según los aspectos antes mencionados para invocar un servicio a través de la PGE:

* Marco legal y técnico
  + SAML
  + Firma digital
  + Certificados digitales
* Intercambio seguro de información
  + WS-Trust
  + WS-Security
  + SSL
* Arquitectura SOA
  + Web Services
  + WS-Addressing

Para facilitar el proceso de invocación, AGESIC ha desarrollado librerías tanto para la plataforma Java como para Microsoft dotNet. De todas formas, se ha detectado que muchos organismos notan cierta dificultad al intentar invocar los servicios, debido principalmente a que por restricciones de seguridad existe alguna complejidad en el proceso. Debido a esto, y para facilitar aún más la tarea de consumo de servicios, AGESIC también desarrolló un aplicativo que hace transparente la complejidad de invocar servicios en la PGE. Este aplicativo es denominado **Conector PGE** o simplemente **Conector**, y funciona como un intermediario entre los clientes propios de la organización y el servicio que se desea consumir.

El Conector se ejecuta dentro de la infraestructura del cliente (el organismo que desea consumir servicios a través de la PGE instala el Conector en su red privada) y básicamente su propósito es actuar como servidor proxy para la invocación a servicios en la PGE: para cada servicio expuesto en la PGE y que se desee consumir se podrá configurar un servicio virtual en el Conector, llamado **“servicio proxy”** que represente al servicio real, el cual ofrece exactamente las mismas operaciones pero sin restricciones de seguridad, siendo el Conector el encargado de aplicar dichas restricciones para luego invocar el servicio final. De esta manera, los clientes finales invocarán al servicio virtual en el Conector, éste tomará los pedidos, enriquecerá el mensaje con lo requerido por la PGE (información de seguridad, autenticación y direccionamiento) e invocará al servicio en la PGE. Una vez obtenido el resultado, el Conector lo retornará al cliente como si fuese el servicio final. Un diagrama de la arquitectura general se ilustra en la Figura 1.

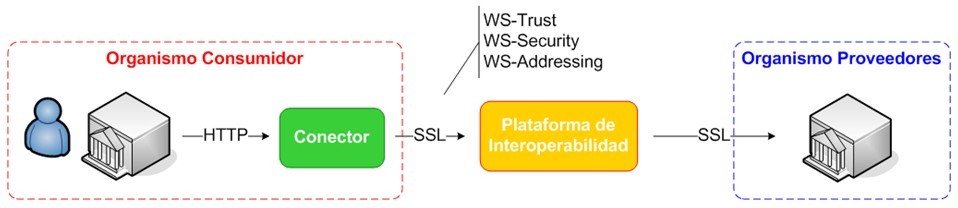


Figura 1 - Diagrama de arquitectura

Para que pueda efectuarse la comunicación, el Conector debe configurarse adecuadamente, y para ello, se cuenta con una consola web de administración. La sección 3, se centra básicamente en explicar cómo configurar una instancia del Conector.

Tener en cuenta que un **conector** representa, en la práctica, a una cierta **operación** dentro de un **servicio**. Si se desea utilizar más de una operación de un mismo servicio, será necesario definir tantos conectores como operaciones requeridas, utilizando el mismo WSDL. Por otra parte, no es necesario crear un conector para una operación de un servicio si dicha operación no será utilizada.

El propósito de este documento es presentar detalladamente cómo configurar el Conector para invocar servicios expuestos en la PGE. Dado que su uso es muy simple, el tutorial se centrará en la configuración de un servicio en particular a modo de ejemplo, llamado Servicio de Timestamp provisto por la PGE, el cual es un servicio devuelve la fecha y hora de la Plataforma de Interoperabilidad.

El resto del documento se organiza de la siguiente manera: en la sección 2 se explica cómo instalar y configurar el Conector, la sección 3 se presenta cómo configurar los servicios proxy (servicios “virtuales”) en el Conector para acceder a los servicios expuestos en la PGE, la sección 4 comenta sobre respaldo y recuperación, y la sección 5 sobre configuración avanzada.

# **Instalación y configuración**

El paquete de instalación del Conector PGE es un archivo de nombre: *conector-3.0.zip* que empaqueta cuatro carpetas:

* apache-tomcat-8.5.23: Incluye el servidor Apache Tomcat versión 8.5.23, en el cual se podrá ejecutar el conector
* wildfly-9.0.2.Final: Incluye el servidor Wildfly versión 9.0.2, en el cual se podrá ejecutar el conector
* jre1.8.0\_151\_linux: Incluye JRE 1.8 para Linux
* jre1.8.0\_151\_win: Incluye JRE 1.8 para Windows

El proceso de instalación del Conector consiste en extraer el archivo *conector-3.0.zip* en el directorio deseado. Debe tenerse en cuenta que la única restricción al elegir el directorio es que **no debe contener espacios** en la ruta. Para el resto del documento se utilizará <conector-pge> para hacer referencia al directorio donde quedó instalado el Conector (por ejemplo */opt/agesic/conector-pge o C:\agesic\conector-pge).*

En la ruta antes mencionada se encontrará el archivo *conector-3.0.zip* así como una carpeta de nombre “*archivos tutorial”* que contiene los archivos que se necesitarán para el desarrollo del tutorial, que deberán ser descargados previamente:

* Timestamp\_wsdl.xml
* Timestamp\_ssl.keystore
* Timestamp\_ssl.truststore
* Timestamp\_organismo.keystore

## **Configuración básica**

El Conector está compuesto de dos componentes: integration y web.

## **Configuración básica componente web**

Para configurar el componente web se deberá abrir el archivo connector-web.properties ubicado en:

* Wildfly: “<conector-pge>/wildfly-9.0.2.Final/standalone/configuration */connector-web.properties”.*
* Tomcat: “<conector-pge>/apache-tomcat-8.5.23/conf*/connector-web.properties”.*

La Figura 2 ilustra las propiedades de este archivo las cuales son explicadas a continuación.



Figura 2: Archivo de configuración módulo web

**userPassword**: Configura la contraseña del usuario *conector*

## **Configuración básica componente Integration**

Para configurar el componente web se deberá abrir el archivo connector.properties ubicado en:

* Wildfly: “<conector-pge>/wildfly-9.0.2.Final/standalone/configuration */connector.properties”.*
* Tomcat: “<conector-pge>/apache-tomcat-8.5.23/conf*/connector.properties”.*

La Figura 3 ilustra las propiedades de este archivo las cuales son explicadas a continuación.

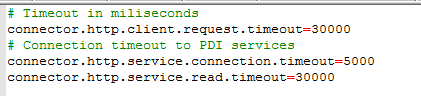


Figura 3 - Archivo de configuración módulo integration

**connector.http.client.request.timeout**: Permite configurar el tiempo máximo de procesamiento de un mensaje recibido por el conector. En caso de que un mensaje recibido no sea proceado antes de este tiempo, se lanzará un mensaje de error al cliente.

**connector.http.service.connection.timeout**: Permite configurar el tiempo máximo de espera de un para establecer la conexión con un servicio de la PDI.

**connector.http.service.read.timeout**: Permite configurar el tiempo máximo de espera para recibir una respuesta de un servicio de la PDI

## **Ejecución**

Luego de instalado y configurado el Conector como se explicó en la sección 2.1, debe iniciarse el servidor de Apache Tomcat o Wildfly.

Para Apache Tomcat, se debe ejecutar el script *startup.sh* (en Linux) o *startup.bat* (en Windows) que se encuentran en la carpeta < *conector- pge>/apache-tomcat-8.5.23/bin/.* Para Wildfly, se ejecuta el script *run.sh* (en Linux) o *run.bat* (en Windows) ubicado en la carpeta *<conector-pge>/wildfly-9.0.2.Final/bin/*.

En caso de Wildfly, debe especificarse el parámetro -b 0.0.0.0 o sino el acceso al conector sólo podrá ser realizado desde el mismo equipo (localhost). Para ejecutar el script mencionado puede especificarse la ruta completa, o cambiar al directorio correspondiente a cada servidor y ejecutarlo directamente:

**Apache Tomcat**

* **Linux**:
  + Opción 1
    - cd <conector-pge>/apache-tomcat-8.5.23/bin/
    - ./startup.sh
  + Opción 2
    - <conector-pge>/apache-tomcat-8.5.23/bin/startup.sh
* **Windows**:
  + Opción 1
    - cd <conector-pge>\apache-tomcat-8.5.23\bin\
    - startup.bat
  + Opción 2
    - <conector-pge>\apache-tomcat-8.5.23\bin\startup.bat

**Wildfly**

* **Linux**:
  + Opción 1
    - cd <conector-pge>/wildfly-9.0.2.Final/bin/
    - ./standalone.sh -b 0.0.0.0
  + Opción 2
    - <conector-pge>/wildfly-9.0.2.Final/bin/standalone.sh -b 0.0.0.0
* **Windows**:
  + Opción 1
    - cd <conector-pge>\wildfly-9.0.2.Final\bin\
    - standalone.bat -b 0.0.0.0
  + Opción 2
    - <conector-pge>\wildfly-9.0.2.Final\bin\standalone.bat -b 0.0.0.0

Una vez iniciado el servidor deberá verse en la consola la información de arranque y ejecución (log) del servidor similar a la mostrada en la Figura 4 y , donde se resalta que los componentes relativos al Conector están instalados, el servidor está en ejecución, y no se visualizan excepciones de ningún tipo.

**Apache Tomcat**

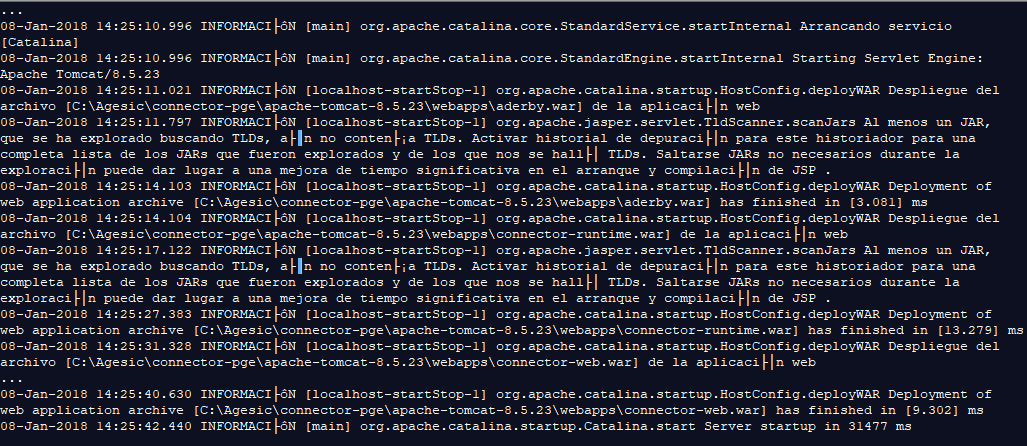


Figura 4 - Inicio En servidor Tomcat

**Wildfly**

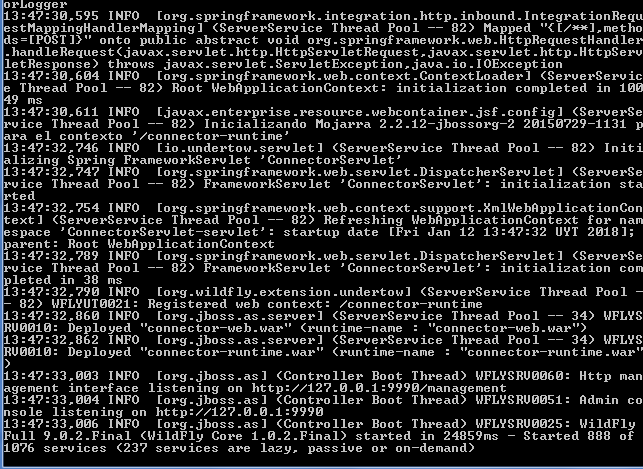


Figura 5 - Inicio de servidor Wildfly

Para confirmar que el conector funciona correctamente se debe acceder a su consola de administración, a través de la URL <http://hostname:9700/connector-web/loginPage>[[1]](#footnote-1)*.* Accediendo a esta URL se deberá ver la interfaz gráfica de la consola de administración tal como se ilustra en la Figura 6.



Figura 6 - Página inicial de la consola de administración

# **Administración de conectores**

En las siguientes secciones se detalla el uso de la consola de administración del Conector para configurar servicios proxy que accedan a servicios publicados en la PGE. Se comenzará creando un servicio proxy para acceder al servicio Timestamp, se ilustrará paso a paso su configuración, y a partir de éste, se ilustrará cómo realizar operaciones de mantenimiento sobre el mismo (edición, eliminación, exportación, etc).



## **Acceso**

Para acceder al Conector una vez instalado y configurado como se especificó en la sección 2.2, se debe acceder a la consola de administración que se presentó en la Figura 6. Se debe proporcionar un usuario y una contraseña; por defecto, y a menos que se modifique, debe especificarse “*conector*” tanto como usuario como contraseña. Por detalles sobre cómo cambiar la contraseña, y las formas de autenticación consultar la sección 2.1.1.

Al acceder a la consola de administración del conector se desplegará una pantalla similar a la que muestra la Figura 7, en la cual se aprecia el listado de conectores (inicialmente vacío) y una lista de selección que permite seleccionar el ambiente sobre el cual se desea trabajar (testing o producción).

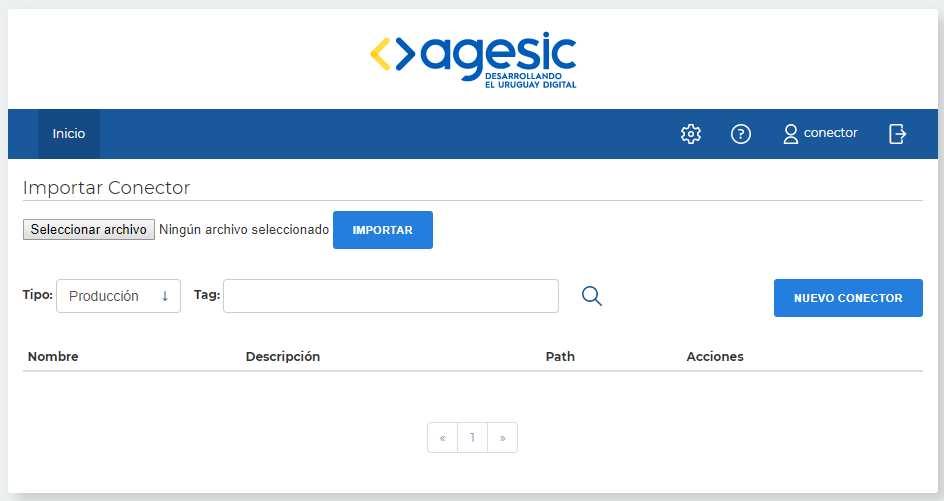


Figura 7 - Listado de conectores para ambiente de producción

Cada línea en el listado representa un **conector** (también llamado **servicio proxy**) a un servicio publicado en la PGE, esto es, un punto de acceso a un servicio en un organismo a través del Conector.

Se puede apreciar que la pantalla tiene una lista de selección (combo), a los efectos de especificar si se desea trabajar en el ambiente de testing o de producción. Esto debe ser tenido en cuenta al momento de crear un conector ya que se deberá especificar si éste invocará a un servicio final en el ambiente de test o en el de producción. En caso de querer crear servicios proxy para un mismo servicio en los dos ambientes a la vez (testing y producción), se deberán crear dos conectores independientes, cada uno en su respectivo ambiente. Para poder visualizar los conectores definidos para los diferentes ambientes, "Testing" o "Producción" bastará con seleccionar el ambiente deseado en el combo Tipo, ubicado en la esquina superior izquierda de la pantalla.

Sobre cada conector definido la consola permite efectuar varias acciones: Visualizar, Editar, Eliminar, obtener el WSDL, y exportar la configuración en formato XML. En la Figura 8 se resaltan los botones de acciones (se encuentran en el orden especificado). Además, la consola de administración permite crear un nuevo conector, e importar un conector a partir de la exportación de otro.

A continuación, se explica cada una de estas acciones.

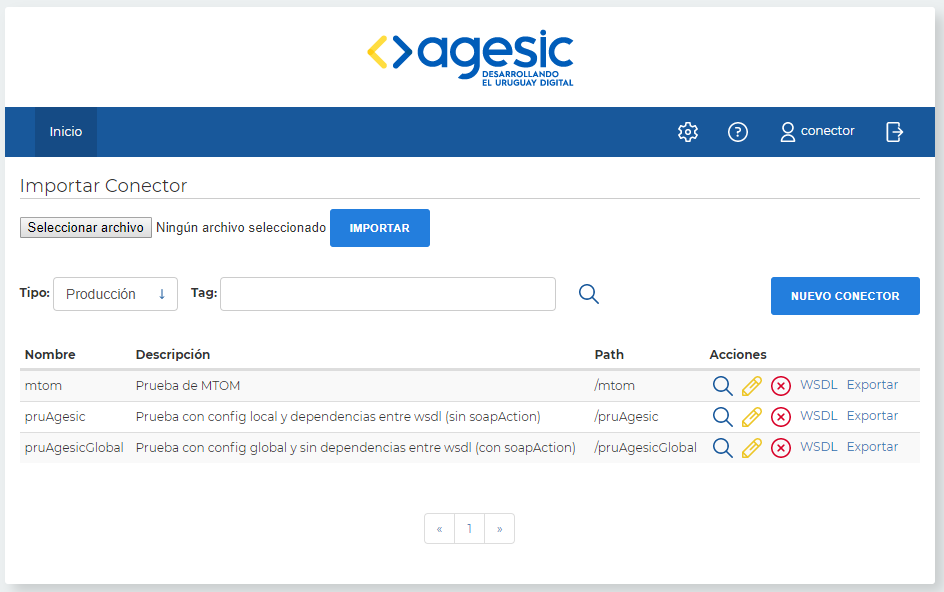


Figura 8 - Funcionalidades para los conectores

## **Crear un nuevo Conector**

Para crear un conector se debe presionar el botón “**NUEVO CONECTOR**” según se ilustra en la Figura 9 y seguir los pasos del formulario.

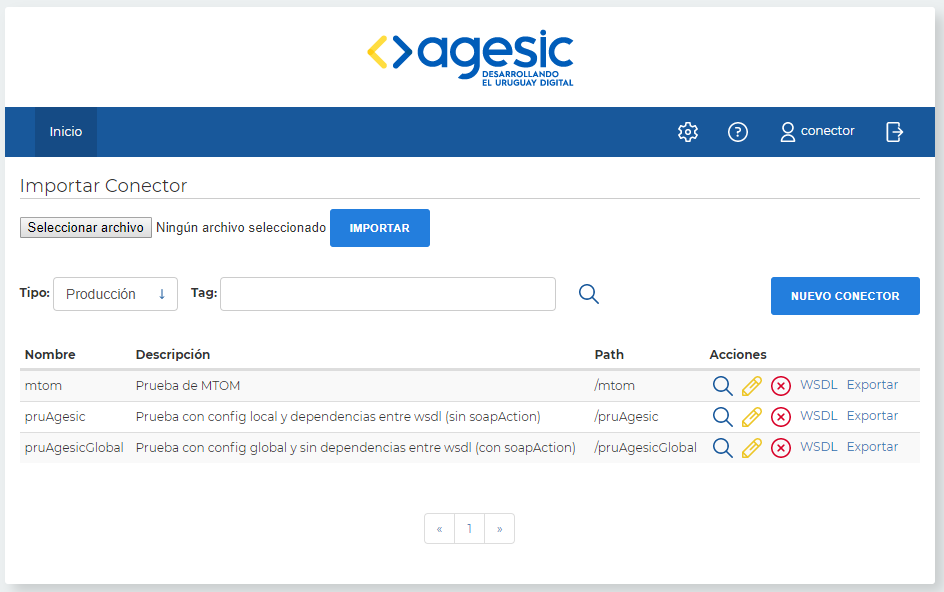


Figura 9 - Crear un nuevo conector

El primer paso, ilustrado en la Figura 10 permite subir el WSDL del servicio en dos modalidades: WSDL unitario o WSDL más XMLSchemas. La opción WSDL unitario es para aquellos Web Services que poseen un único WSDL y que su XMLSchema se encuentra embebido en este archivo. La segunda opción aplica para aquellos Web Services donde el WSDL importa uno o más XMLSchemas. En esta modalidad se debe empaquetar todos los archivos en un archivo Zip. Por más detalles sobre cómo estructurar este archivo para que sea correctamente interpretado por el Conector, ver Anexo I – Creación de archivo Zip.

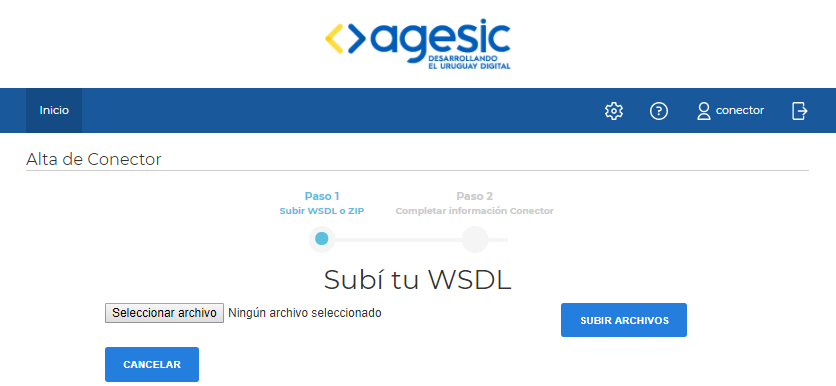


Figura 10 - Paso 1 para la creación de un conector

Como segundo paso, se deben ingresar los datos de los campos solicitados como se muestra en la Figura 11 y presionar el botón Alta. Los campos señalados con \* son obligatorios, por lo que se desplegará un error en caso que falten.

**Nota:** la combinación de los campos "Nombre" y "Tipo" debe ser única en el conector, así como también la combinación de los campos "Path" y "Tipo". Por ejemplo, no pueden existir dos conectores en el ambiente de testing llamados ambos “Servicio1”, aunque sí podrían existir dos conectores llamados “Servicio1”, si uno es creado en el ambiente de testing y el otro en el ambiente de producción.

A continuación, se detalla la semántica de cada uno de los campos y el valor correspondiente que deben tomar para este caso particular del servicio Timestamp (como se mencionó anteriormente, el conector que se creará en este tutorial, como ejemplo, será un servicio proxy al servicio Timestamp, que devuelve la fecha y hora actual de la PGE).

Figura 11 - Paso 2 para la creación de un conector

* **Nombre**: corresponde al nombre del conector. Se usa como identificación interna del conector y en combinación con el campo Tipo deben ser únicos. Este valor es elegido por el usuario e independiente del servicio. En el caso del ejemplo se ingresará “*Timestamp”*.
* **Tipo**: identifica al ambiente en el cual se desea trabajar; las opciones disponibles son “Testing” y “Producción”. En el caso del ejemplo se selecciona “*Testing”* dado que se desea consumir un servicio del ambiente de testing.
* **Descripción**: permite ingresar alguna información descriptiva del servicio, sin uso operativo. En el caso del ejemplo se ingresará “*Servicio Timestamp de Test”*.
* **Path**: corresponde a la parte de la url en donde quedará disponible el WSDL del servicio proxy que se está dando de alta. Los WSDLs quedaran disponibles en una URL compuesta de tres partes: *server\_host*, *server\_port* y *path*. El valor de server\_host es el nombre del equipo donde se ejecuta el Conector, el server\_port corresponde al puerto por el que atiende el Conector (ver Anexo III – Puertos del conector), y path es el valor que se coloca en este campo. De esta manera, la URL en donde se desplegará el WSDL del servicio virtual a construir será de la siguiente forma: [**http://server\_host:server\_port/pat**](http://server_host:server_port/path)**h?wsdl**. El valor de este atributo puede ser elegido libremente por el usuario, teniendo en cuenta que la combinación de los atributos Tipo y Path debe ser única. Para el caso del ejemplo se definirá el path “*/Test/Timestamp”* (con esto, una vez terminada la creación del conector el WSDL quedará disponible en la URL [*http://server\_host:server\_port/Test/Timestamp).*](http://server_host:server_port/Test/Timestamp)En caso que se active la opción de uso de https, el protocolo de comunicación será https en lugar de http.
* **Url**: dirección física del servicio en la PGE. Este campo es autocompletado con la información del WSDL subido en el paso 1 de este tutorial (ver service/port/address en el WSDL) y puede ser modificado en este paso.
* **Wsa:To**: este dato será agregado por el Conector como encabezado (header) del mensaje SOAP que se enviará a la PGE. Corresponde a un endpoint lógico, que permite a la PGE determinar cuál es el servicio concreto que se desea invocar. Este dato debe ser proporcionado por AGESIC cuando se brinda el formulario con la información necesaria para invocar al servicio. Para el caso del ejemplo, se colocará el siguiente valor [*http://testservicios.pge.red.uy/timestamp*](http://testservicios.pge.red.uy/timestamp).
* **Username**: es el usuario dentro del organismo que desea acceder al servicio final; este dato es utilizado para la firma del mensaje SOAP. Debe ser proporcionado por el organismo consumidor. Cabe mencionar que la PGE no realiza ninguna validación sobre este campo, pero es necesario colocar un valor que tenga significado para el organismo a efectos de servir para la auditoría de las transacciones. En este caso se especifica el valor *jperez*.
* **Organismo**: es el nombre que identifica al organismo consumidor; este dato es utilizado también para la firma del mensaje SOAP, y tiene el mismo propósito que el campo anterior. En este caso se especifica el valor *Agesic*.
* **Ingresar credenciales de Username Token**: este checkbox permite habilitar el ingreso de usuario y contraseña, en caso que el servicio requiere WS-Security/Username Token. Luego de activar este checkbox, se habilitarán los campos de texto para ingresar esta información. Estos campos de texto se encuentran debajo del último checkbox (Usar configuración STS local)
* **Habilitar configuración local**: este checkbox permite ingresar una configuración específica a este servicio para la configuración de keystores y truststores. Una vez habilitada esta opción se deben ingresar los siguientes campos:
* **Alias del Keystore de Organismo**: el valor de este campo determina el certificado que se desea utilizar, el cual debe estar dentro del almacén de claves del organismo. El keystore de organismo se encuentra en el archivo cargado en el campo el Keystore de Organismo. En el ejemplo se especifica el valor *testing.hg.red.uy.*
* **Password del Keystore de Organismo**: este campo contiene la contraseña del almacén de claves especificado por el campo Keystore de Organismo. En el ejemplo se especifica el valor *conector*.
* **Password Keystore SSL**: este campo contiene la contraseña del almacén de claves especificado por el campo Keystore de SSL. En el ejemplo se especifica *conector.*
* **Password Truststore**: este campo contiene la contraseña del almacén de claves especificado por el campo Truststore de SSL. En el ejemplo se especifica *conector*.
* **Keystore de Organismo**: este campo permite ingresar la ruta al keystore de organismo utilizado para la firma del mensaje SOAP. En el ejemplo, se carga el archivo *Timestamp\_organismo.keystore*.
* **Keystore de SSL**: este campo permite ingresar la ruta al keystore utilizado para la comunicación HTTPS. En el caso del ejemplo, se carga el archivo *Timestamp\_ssl.keystore*.
* **Truststore de SSL**: este campo permite ingresar la ruta al truststore utilizado para la comunicación HTTPS con autenticación mutua. En el caso del ejemplo, se carga el archivo *Timestamp\_ssl.truststore*.
* **Habilitar caché de tokens:** este checkbox permite habilitar y deshabilitar el caché de tokens SAML. Cuando esta opción está habilitada se crea un caché de tokens SAML donde se almacenarán los tokens SAML solicitados al STS. En caso de que exista un token SAML válido en el caché, no se solicitará uno nuevo al STS, si no, que se utilizará este token del caché. En casos donde el token esté vencido o no sea válido, se solicitará un nuevo token al STS y se almacenará este nuevo token en el caché sustituyendo al antigüo token. Esta opción proporciona una mejora considerable en la performance de la invocación a los servicios de la PGE en casos donde haya una carga importante sobre el servicio. La duración estándar de los tokens SAML es de 15 minutos.
* **Publicar por https:** Por defecto, todos los pedidos a un servicio proxy reciben los pedidos utilizando el protocolo http en los puertos 9800 (testing) y 9700 (producción). Este checkbox permite modificar este comportamiento y configurar el **servicio proxy** para que sólo reciba pedidos por https en los puertos 8443 (testing) y 8553 (producción).
* **Usar configuración STS local:** La url del STS a utilizar se puede configurar en la Configuración global del conector, donde todos los **servicios proxy** utilizarán esta url, pero también es posible ingresar una configuración específica a un **servicio proxy** habilitando el este checkbox. En este caso se desplegará un campo de texto donde ingresar la url del STS. El resto de los **servicios proxy** configurados en el conector seguirá utilizando la url definida en la Configuración global.
* **TAG:** permite ingresar un conjunto de palabras clave para luego poder buscar este conector por estas palabras.

La Figura 12 muestra los campos ya cargados con los datos apropiados.

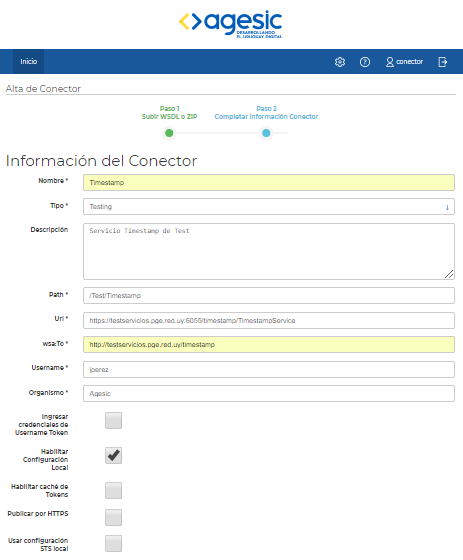


Figura 12 - Conector configurado

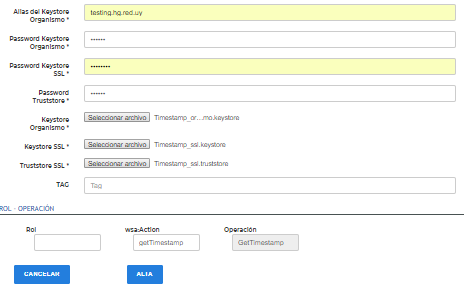


Figura 13 - Conector configurado (parte 2)

Una vez configurada la información general de un servicio proxy, es necesario ingresar la configuración de seguridad por operación. Los campos que deberán cargarse son *Rol* y *wsa:Action* para cada una de las operaciones del WSDL. A continuación, se explica el significado de cada uno de ellos, así como el valor que deben tomar para la configuración del conector para el servicio de Timestamp.

* **Rol**: es utilizado para autorizar el acceso a los servicios, y debe ser proporcionado por AGESIC para cada una de las operaciones que se desea invocar. En el caso del ejemplo, se ingresará el valor *OU=TEST\_TUTORIAL,O=TEST\_PE*.
* **Wsa:Action**: determina el método al que corresponde la operación en el servicio final. El valor apropiado para cada operación se obtiene del WSDL: es valor del atributo *soapAction* del tag *operation*. En el caso del ejemplo, se debe especificar el valor <http://test.pge.red.uy/timestamp/GetTimestamp> (confirmarlo observando el WSDL). Este valor ya viene precargado luego de subir el WSDL para aquellos WSDLs que cuenten con un valor distinto de vacío. En caso que no cuente y sea vacío, se deberá completarse de la siguiente manera:
  + Testing: <http://testservicios.pge.red.uy/nombreServicio/nombreMetodo>
  + Producción: <http://servicios.pge.red.uy/nombreServicio/nombreMetodo>

Este dato es importante para el registrar qué método consume el organismo, el cual queda registrado en la aplicación de Reportes de la Plataforma.

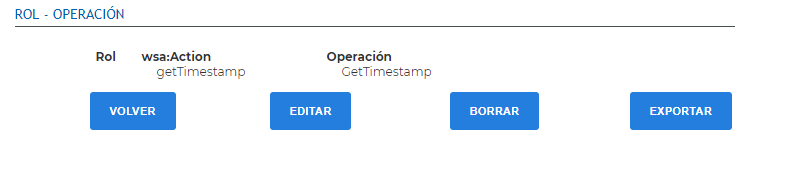


Figura 14 - Cargar roles por operación

Una vez completada toda la información del formulario, se presiona el botón *ALTA*, y se ha terminado de crear el conector. Tras esto, se observará la misma pantalla de la Figura 10, mostrando los datos del conector recientemente creado.

En este punto, se ha creado satisfactoriamente un conector que permite invocar al servicio Timestamp provisto por AGESIC a través de la PGE. En las siguientes secciones se mostrará cómo realizar operaciones de mantenimiento sobre este conector.

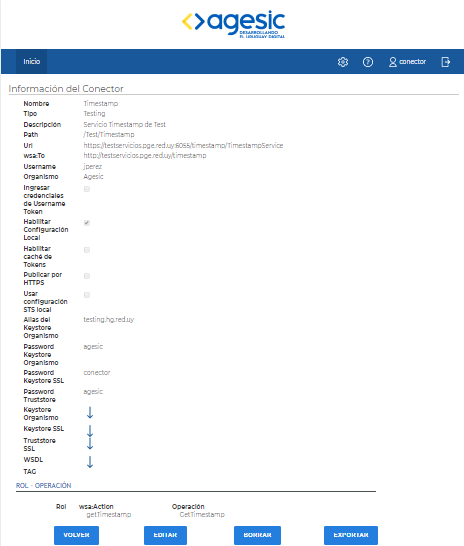


Figura 15 - Información completa del conector

## **Ver Conector**

Una vez creado un conector podrán visualizarse sus datos haciendo clic sobre la acción “Ver Conector” que se representa con un ícono similar a una lupa, como se muestra en la Figura 16. De esta manera, podrán verse la totalidad de los campos del conector. Se permite, además, la posibilidad de borrar o editar el conector presionando el botón correspondiente que aparece en el pie de la pantalla (Editar, Borrar). Un ejemplo de la pantalla que se despliega luego seleccionar la acción “Ver Conector” se ilustra en la Figura 11.

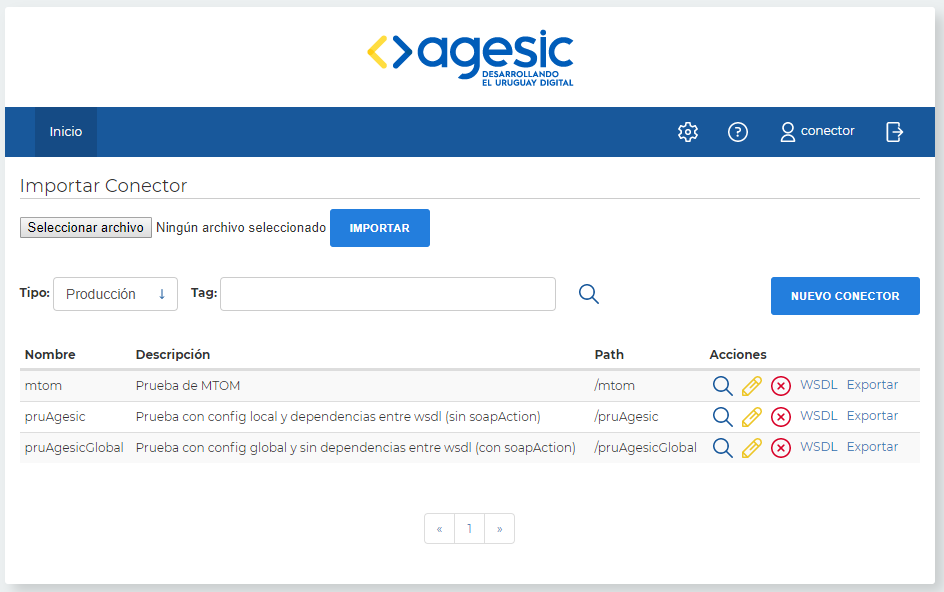


Figura 16 - Ver conector

De esta manera, podrán verse la información completa del conector, permitiendo la posibilidad de borrar o editar el conector presionando el botón correspondiente que aparece en el pie de la pantalla (Editar, Borrar). Asimismo, desde esta pantalla es posible descargar los kesystores y truststores asociados a un conector, si se optó por una configuración local. Desde esta misma pantalla también es posible descargar el WSDL del conector. La pantalla de visualización de un Conector se ilustra en la Figura 17.

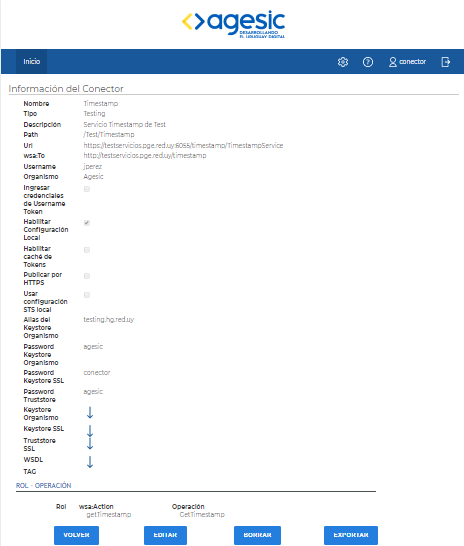


Figura 17 - Ver conector (parte 2)

## **Editar un conector existente**

Desde el listado de conectores se puede acceder a una página para editar cada uno de ellos; para hacerlo, se debe presionar el botón Editar, donde se desplegará un formulario para actualizar el WSDL del servicio. En esta pantalla, se podrá optar por mantener el archivo WSDL o ZIP subidos anteriormente, o cargar nuevos. Luego, se procede a editar el resto de los campos, como se ilustra en la Figura 18.

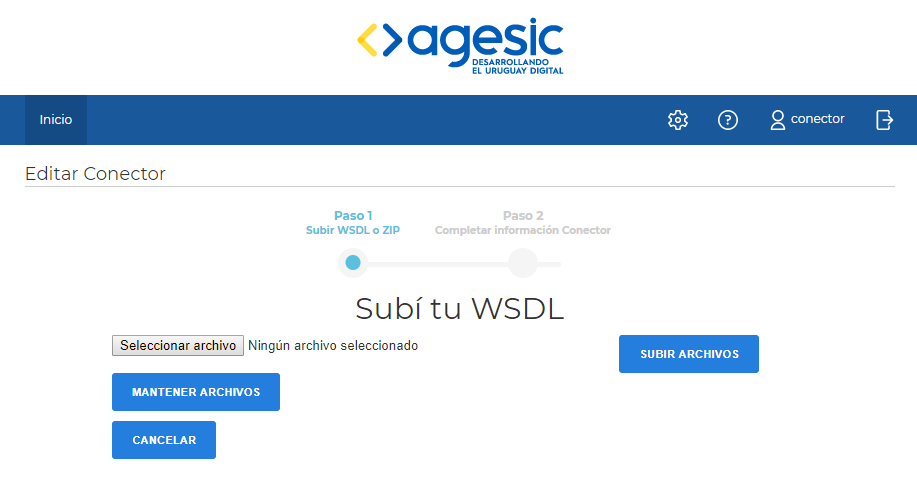


Figura 18 - editar un conector: modificar wsdl

Luego, en la siguiente pantalla podrán editarse todos los datos del conector, salvo Nombre, Tipo y URL, según se muestra en la Figura 19.

En casos donde se haya seleccionado la configuración local de los keystores y truststores, es posible descargar o visualizar la información que contienen. La Figura 20 muestra un ejemplo de visualización del contenido de un keystore.

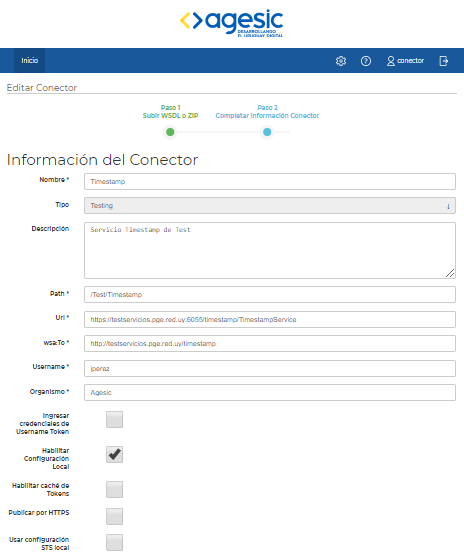


Figura 19 - Modificar información del conector

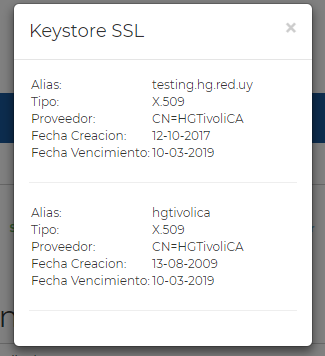


Figura 20 - Visualización de información de keystores/Truststores

## **Eliminar un conector existente**

Se podrá eliminar un conector desde el listado de conectores haciendo clic sobre el botón Borrar Conector. En todos los casos que se desee borrar un conector, se pedirá confirmación tal como se ilustra en la Figura 21.

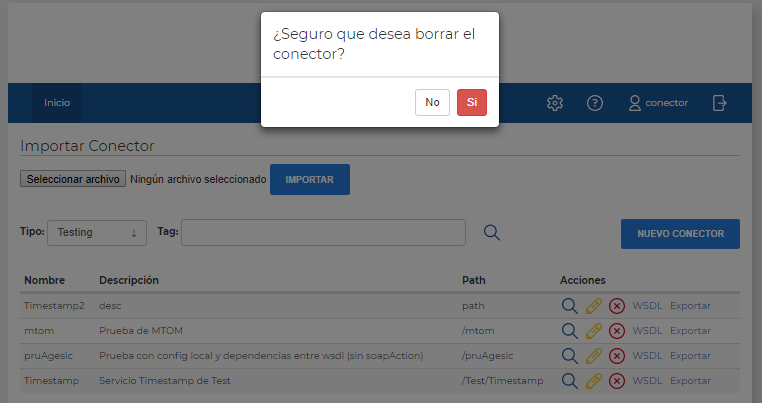


Figura 21 - Eliminar un conector

## **Ver WSDL**

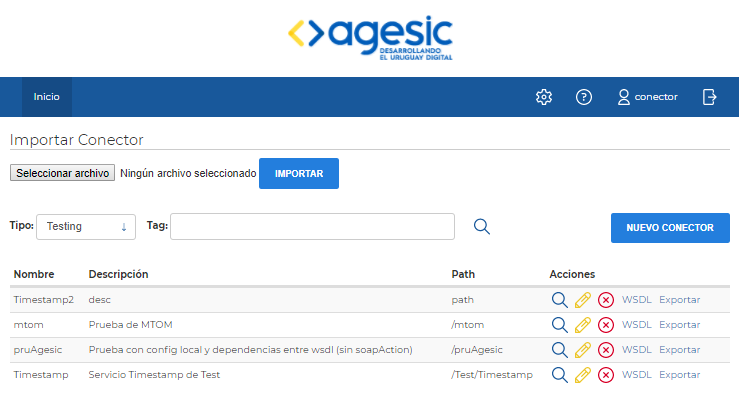
Para visualizar el WSDL correspondiente a un conector se debe hacer clic sobre el link WSDL asociado a un conector, como se muestra en la Figura 22.

Figura 22 - Visualizar WSDL

En la Figura 23 se muestra la visualización del WSDL generado para el ejemplo. Dicho WSDL es el que se debe utilizar para invocar al servicio publicado en la plataforma.

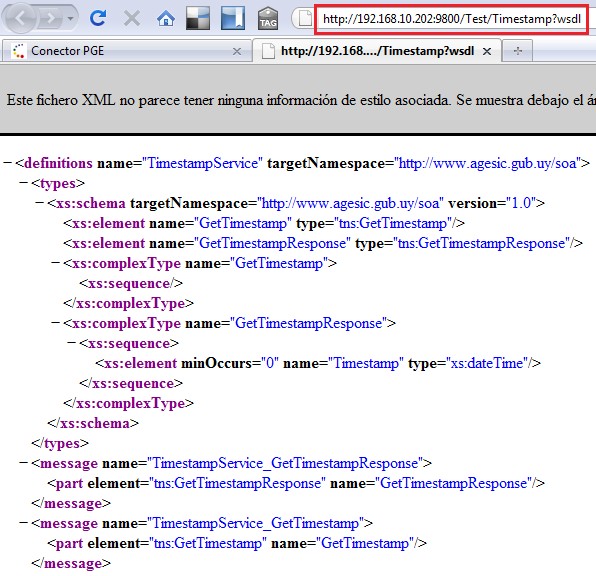


Figura 23 - Visualización del WSDL del servicio Timestamp

## **Exportación de un conector**

La exportación de la configuración de los conectores permite realizar copias de seguridad de ellos o la clonación de la configuración para luego importarla en otra instalación del Conector.

Para exportar un conector a un archivo (en formato XML) se debe hacer clic en el link EXPORTAR asociado al conector, tal como se ilustra en la Figura 24. También es posible realizar esta misma acción desde la vista de un conector específico, tal como se muestra en la Figura 25

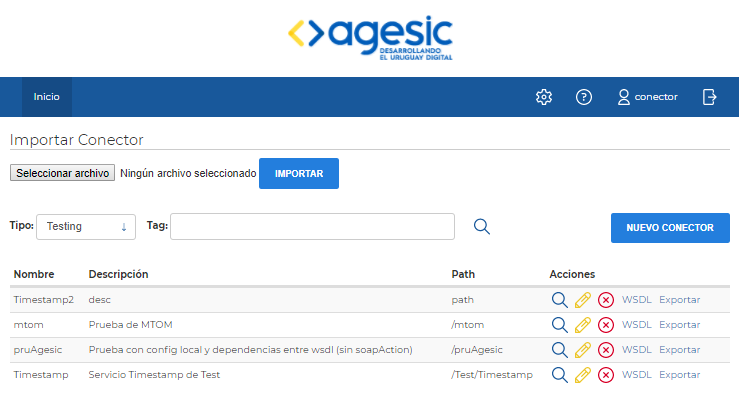


Figura 24 - Exportar configuración desde el listado de conectores

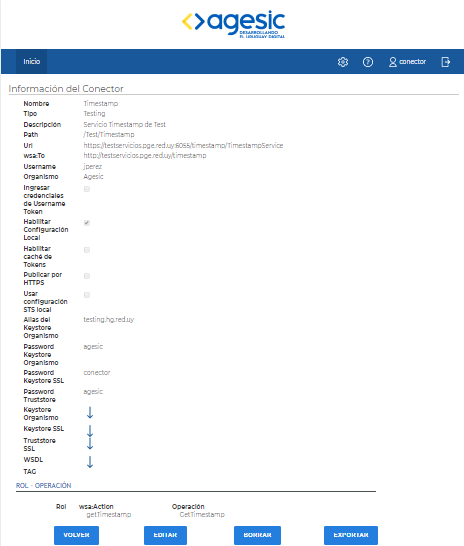


Figura 25 - Exportar configuración desde un conector

La Figura 26 se ilustra el XML generado en la exportación.



Figura 26 - Exportación de configuración para el servicio de Timestamp

## **Importación de Conector**

Una vez exportado un conector (ver sección 3.7) es posible importar su configuración y así evitar ingresarla manualmente. Para realizar esta tarea, en la página de inicio existe un área de importación en donde se puede seleccionar un archivo XML exportado previamente. Es posible importar conectores de la versión 3.0 así como también de la versión 2.0. En la figura 18 se ilustra la sección de importación.

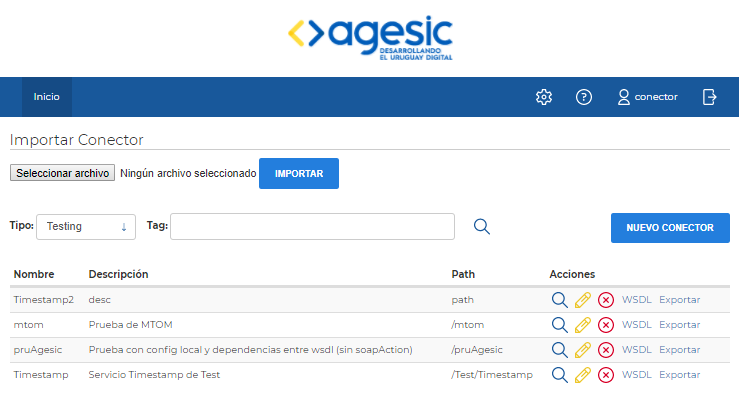


Figura 27 - Importación de un conector

Debe tenerse en cuenta al importar un conector a partir de un archivo que la combinación "Nombre" - "Tipo" tiene que ser única en el sistema, así como también la combinación "Path" – "Tipo". Por lo tanto, en caso de que exista un conector en el sistema que presente algún conflicto se desplegará una validación que impedirá la importación.

## **Remplazo de keystores**

Los keystores que fueron utilizados para crear los conectores contienen certificados digitales. Estos certificados tienen un período de validez, fuera del cual no serán aceptados por la PGE de AGESIC. En caso de vencimiento de uno de los certificados dentro de alguno de los keystores, será necesario remplazar el keystore. Para hacerlo, se debe realizar el siguiente procedimiento:

1. Acceder a la consola del conector.
2. Exportar del servicio, como respaldo del mismo (en caso de algún problema, se puede volver a crear el servicio a partir de este archivo). Para esto, hacer clic sobre el link "Exportar" en la fila del servicio.
3. Editar el servicio, haciendo clic en el botón "Editar conector" (el segundo botón, junto a la lupa).
4. Especificar la ubicación del keystore que se desea actualizar; para esto, hacer clic en el botón "Examinar" correspondiente al keystore que se desea actualizar (“Keystore de organismo”, “Keystore SSL”, “Truststore SSL”), navegar hasta el archivo que se desea cargar, seleccionarlo y hacer clic en el botón "Abrir".
5. Hacer clic en el botón "Actualizar" al final de la página.

Repetir este procedimiento para cada servicio al cual se deba actualizar algún keystore. Recordar que en el ambiente de testing, los keystores de organismo y SSL pueden ser el mismo, por lo que seguramente, en caso de remplazar uno de ellos, deba remplazarse el otro también.

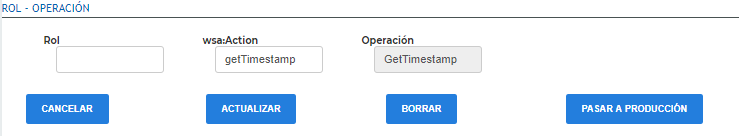
## **Pasaje a producción**

Luego de tener el servicio funcionando correctamente en el ambiente de Testing, es posible realizar un pasaje semi automático al ambiente de producción. Esta funcionalidad permite crear un conector en el ambiente de producción con los mismos datos que se encuentran en el ambiente de Testing, salvo los listados a continuación:

* Keystores
* Contraseñas
* Roles
* URL destino
* Alias
* Usuario y contraseña de WS-Security/Username Token

Para realizar el pasaje a producción, se deben seguir los siguientes pasos:

1. Desde la página de inicio, filtrar los conectores por tipo Testing
2. Seleccionar el conector para pasar a producción
3. Editar el conector haciendo clic en “Editar conector”
4. Se despliega el formulario, en el Paso 1, hacer clic en el botón “Mantener archivos”
5. En el paso 2, hacer clic en el botón “Pasar a producción”. Ver Figura 19.



*Figura 19: Pasaje a producción*

## **Configuración global**

El sistema permite ingresar una configuración global a todos los conectores dados de alta, para así no tener que ingresar esta información que generalmente es la misma para todos los conectores de un mismo ambiente. En casos donde se requiere tener un conector con una configuración local, diferente a la configuración global, se deben seguir los pasos descritos en la sección 3.2. Para acceder a esta funcionalidad, se debe hacer clic en el botón de Configuración Global, situado en la barra de tareas según se muestra en la Figura 28.

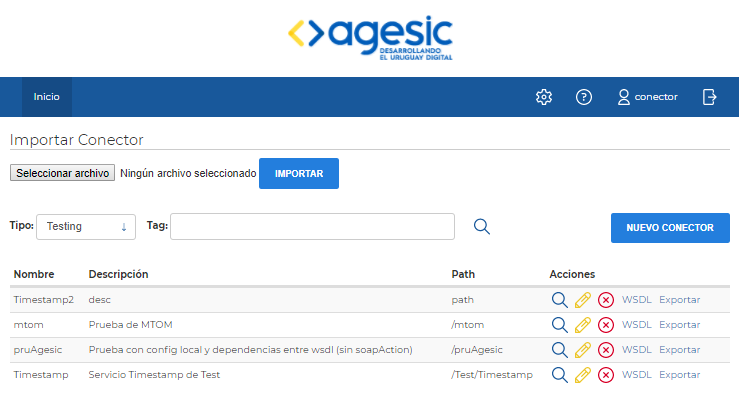


Figura 28 - Configuración global

Tal como se muestra en la Figura 29, desde esta funcionalidad es posible configurar las siguientes opciones:

* Alias del Keystore del Organismo: representa el alias del certificado dentro del keystore de organismo
* Password Keystore del Organismo: contraseña del keystores del organismo
* Password Keystore SSL: contraseña del keystore para la comunicación ssl
* Password Truststore: contraseña del truststore
* Keystore Organismo: archivo keystore donde se encuentra el certificado de persona jurídica del organismo
* Keystore SSL: archivo keystore donde se encuentra el certificado SSL emitido por la PGE para mantener una comunicación segurda entre el conector y la PGE
* Truststore SSL: archivo truststore donde se encuentra la clave pública del certificado ssl de la PGE
* Tipo de token: indica el tipo de token debe configurarse para permitir la invocación de servicios. En el caso del ambiente de testing, el valor de este campo debe ser *urn:tokensimple,* mientras que en el caso del ambiente de producción debe ser *urn:std15*. En este ejemplo, como se desea consumir un servicio del ambiente de testing, se utilizará el valor *urn:tokensimple.*
* URL STS Global: la url del STS de Testing es <https://testservicios.pge.red.uy:6051/TrustServer/SecurityTokenServiceProtected>

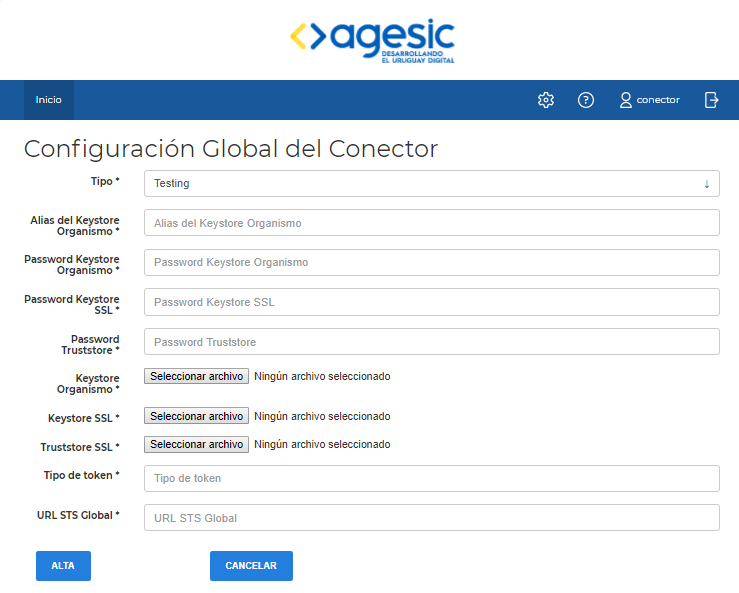


Figura 29 - Configuración global

# **Respaldo y recuperación**

Para realizar el respaldo del conector, se debe realizar una copia de seguridad de los siguientes artefactos:

* Base de datos Derby: El conector cuenta con una base de datos Derby cuyos archivos se encuentran en:
  + Apache Tomcat: *</conector-pge>/apache-tomcat-8.5.23/derby/*
  + Wildfly: *</conector-pge>/wildfly-9.0.2.Final/derby/*
* Archivos keystores, truststores y WSDLs
  + Apache Tomcat: carpeta *</conector-pge>/apache-tomcat-8.5.23/connector*
  + Wildfly: carpeta *</conector-pge>/wildfly-9.0.2.Final/standalone/connector*
* Configuración local de cada componente: ver sección 2.1
* Configuración de puertos:
  + Apache Tomcat: *</conector-pge>/apache-tomcat-8.5.23/conf/server.xml*
  + Wildfly:

*</conector-pge>/wildfly-9.0.2.Final/standalone/configuration/standalone.xml*

Por lo tanto, el respaldo consiste en realizar una copia de seguridad de estos artefactos según el servidor se esté utilizando.

La recuperación consiste en sustituir estos archivos por la copia previamente respaldada. Debe tenerse en cuenta que para realizar las maniobras de respaldo y recuperación, el servidor de aplicaciones (Apache Tomcat o Wildfly) debe estar detenido, ya que en otro caso podrían encontrase inconsistencias.

# **Anexo I – Creación de archivo Zip**

En caso que el WSDL del servicio no sea un único archivo y este importe XMLSchemas, se debe armar un archivo ZIP que contenga el WSDL y los XMLSchemas asociados con la siguiente estructura:

* Archivo Zip
  + Archivo WSDL
  + Archivo XMLSchema 1
  + …
  + Archivo XMLSchema N

Por otro lado, deben existir las siguientes restricciones en los archivos:

* WSDL: si un archivo WSDL importa un archivo XMLSchema, el valor del atributo schemaLocation (que indica la ubicación del XMLSchema) debe ser una dirección relativa a la ubicación del WSDL. Por ejemplo, si el WSDL es un archivo hermano del archivo XMLSchema, el valor de schema location debe ser el nombre del archivo XMLSchema, similar a como se muestra en la Figura 30. En este ejemplo, el nombre del archivo XMLSchema es *WsDebesNeteosBSEService.xsd8.xsd*

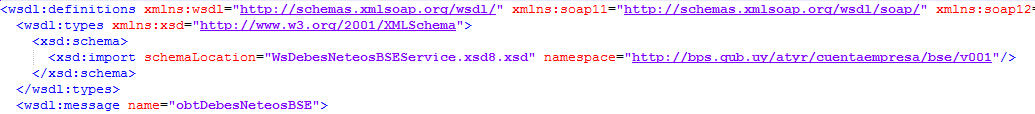


Figura 30 - Ejemplo de importación de XMLSchema dentro de un WSDL

* XMLSchema: si a su vez, un XMLSchema importa otro archivo XMLSchema, se debe seguir un procedimiento similar al anterior. En la sentencia de importación, el valor schemaLocation debe indicar la ubicación del archivo XMLSchema a importar. Esta ubicación debe ser una ruta relativa a la ubicación del XMLSchema importador, tal como se muestra en la Figura 31. En este ejemplo, el archivo XMLSchema importa otros dos archivos: WsDebesNeteosBSEService.xsd7.xsd y WsDebesNeteosBSEService.xsd5.xsd.

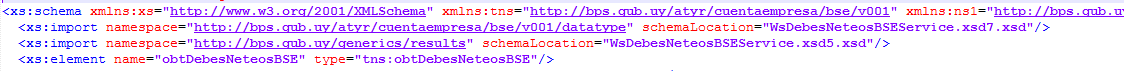


Figura 31 - Ejemplo de importación de XMLSchema dentro de otro XMLSchema

# **Anexos II – Librerías Java**

|  |  |
| --- | --- |
| Librería | Versión |
| Spring Integration | 4.3.2.RELEASE |
| Spring Framework | 4.3.12.RELEASE |
| Spring web | 4.3.12.RELEASE |
| Spring xml | 2.4.2.RELEASE |
| Spring data | 1.11.8.RELEASE |
| Spring security | 4.2.3.RELEASE |
| Derby | 10.13.1.1 |
| Open SAML | 2.6.4 |
| wss4j | 1.6.19 |
| jlibs-xml | 2.2.1 |
| jlibs-xmldog | 2.2.1 |
| Zip4j | 1.3.2 |
| Commons-io | 2.6 |
| Hibernate | 5.2.1.Final |

Tabla 1 - Principales librerías para el desarrollo del conector

# **Anexo III – Puertos del conector**

El conector (en sus dos versiones Tomcat y Wildfly) está configurado para utilizar los siguientes puertos:

* 9700: puerto http para el ambiente de Testing
* 9800: puerto http para el ambiente de Producción
* 8553: puerto https para el ambiente de Testing
* 8443: puerto https para el ambiente de Producción

**Configuración Apache Tomcat**

En Tomcat, para modificar estos puertos se debe ingresar al archivo </conector-pge>/apache-tomcat-8.5.23/conf/server.xml, como se muestra en la Figura 32 y Figura 33.

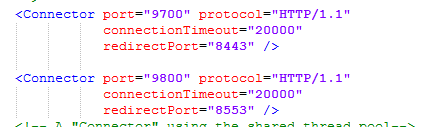


Figura 32 - Configuración de puertos http en Tomcat

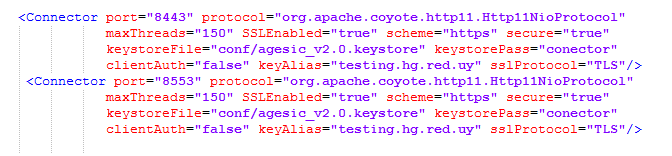


Figura 33 - Configuración de puertos https en Tomcat

Para los puertos https debe configurarse adicionalmente los keystores a utilizar.

**Configuración Wildfly**

En Wildfly, para modificar estos puertos se debe ingresar al archivo *</conector-pge>/wildfly-9.0.2.Final/standalone/configuration/standalone.xml,* como se muestra en la Figura 34.

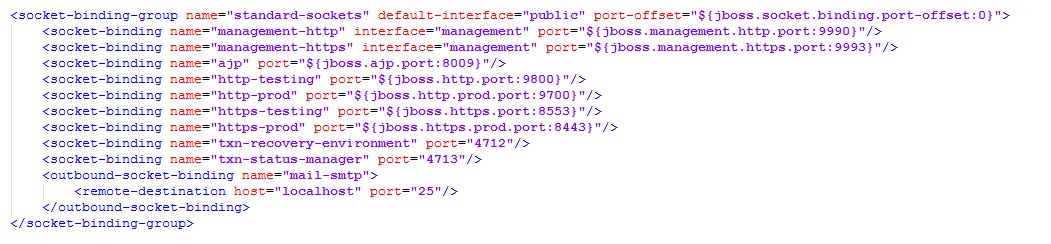
****

Figura 34 - Configuración de puertos http y https en Wildfly

Por otro lado, para configurar los keystores a usar en los puertos ssl se debe modificar la configuración de la Figura 35 y Figura 36.

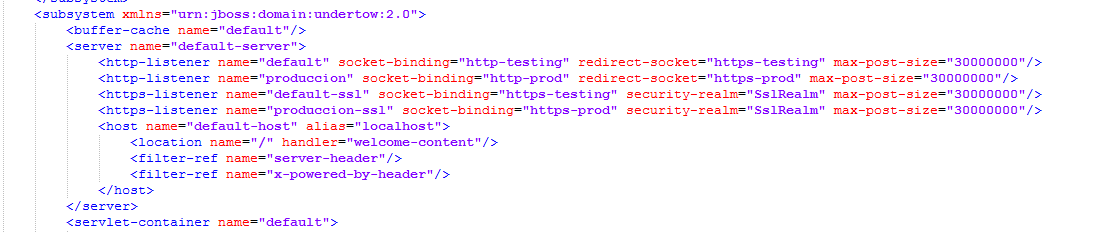


Figura 35 - Configuración de los REALM de SSL

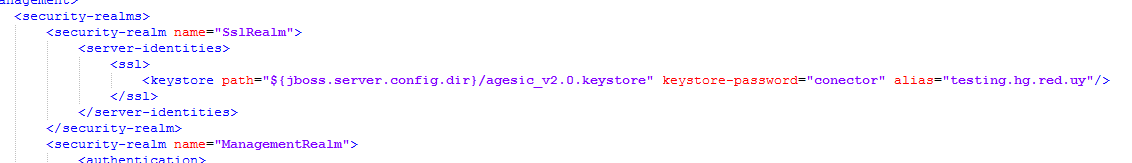


Figura 36 - Configuración de los keystores asociados al REALM de SSL

1. Sustituir *hostname* por ip donde está instalado el servidor. [↑](#footnote-ref-1)