



Plataforma de Interoperabilidad

# Tutorial de configuración del Conector de la PGE

## Control de Cambios

Fecha	Versión	Descripción	Autor	Aprobado Por
	1.0	Versión inicial		
2015	2.0			

Nombre actual del archivo: AGESIC-Plataforma-Tutorial-ConectorPGE-v02-00.odt

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

## Plataforma de Interoperabilidad

Este documento ha sido elaborado por AGESIC (Agencia para el Desarrollo del Gobierno de Gestión Electrónica y la Sociedad de la Información y el Conocimiento)

Usted es libre de copiar, distribuir, comunicar y difundir públicamente este documento así como hacer obras derivadas, siempre y cuando tengan en cuenta citar la obra de forma específica y no utilizar esta obra para fines comerciales. Toda obra derivada de esta deberá ser generada con estas mismas condiciones.

# Índice de contenido

<a href="#">1</a>	<a href="#">Introducción</a>	<a href="#">5</a>
<a href="#">2</a>	<a href="#">Instalación y configuración</a>	<a href="#">6</a>
<a href="#">2.1</a>	<a href="#">Configuración básica</a>	<a href="#">6</a>
<a href="#">2.2</a>	<a href="#">Configuración como servicio</a>	<a href="#">7</a>
<a href="#">2.3</a>	<a href="#">Ejecución</a>	<a href="#">7</a>
<a href="#">3</a>	<a href="#">Administración de Conectores</a>	<a href="#">10</a>
<a href="#">3.1</a>	<a href="#">Acceso</a>	<a href="#">10</a>
<a href="#">3.2</a>	<a href="#">Crear un nuevo Conector</a>	<a href="#">11</a>
<a href="#">3.3</a>	<a href="#">Ver Conector</a>	<a href="#">16</a>

# 1 1. Introducción

AGESIC desarrolló un aplicativo que hace transparente la complejidad de invocar servicios en la PGE, denominado **Conector PGE** o simplemente **Conector**. 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 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. 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 del Conector PGE

El propósito de este documento es presentar cómo instalar y 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 que devuelve la fecha y hora de la Plataforma de Interoperabilidad.

El resto del documento se organiza de la siguiente manera:

- En la sección se explica cómo instalar y configurar el Conector,
- En la sección se presenta cómo configurar los servicios proxy (servicios “virtuales”) en el Conector para acceder a los servicios expuestos en la PGE.

## 2. Instalación y configuración

El paquete de instalación del Conector PGE es un archivo de nombre: *conector.zip* que empaqueta dos carpetas, *java* y *jboss-as*. La carpeta *java* incluye el JDK 1.6 para Windows y Linux. La carpeta *jboss-as* incluye el servidor JBoss AS versión 5.1.0.GA en el cual ejecutará el conector. El archivo *conector.zip* puede ser descargado del FTP de AGESIC, en

<sftp.agesic.gub.uy/E-Learning/ConectorPGE/conector.zip>.

El proceso de instalación del Conector consiste en extraer el archivo *conector.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*).

**IMPORTANTE:** Como el conector utiliza el disco para guardar archivos, será necesario asignarle permisos de escritura a toda la carpeta del conector para el usuario que corre el jboss donde está alojada la aplicación.

### Configuración básica

Para configurar el Conector se deberá abrir el archivo *connector-pge.properties* ubicado en “<conector-pge>/jboss-as/server/default/conf/connector-pge/connector-pge.properties”. La Figura 2 ilustra las propiedades de este archivo las cuales son explicadas a continuación.

```
#BASIC_PATH=/opt/Agesic/connector
HOST=192.168.10.202
PORT_TEST=9800
PORT_PROD=9700
stsURL.production=https://servicios.pge.red.uy:10001/TrustServer/SecurityTokenServiceProtected
stsURL.test=https://testservicios.pge.red.uy:6051/TrustServer/SecurityTokenServiceProtected
```

Figura 2: Archivo de configuración del Conector PGE

- **BASIC\_PATH:** Configura la ruta en el sistema de archivos donde se van a guardar los archivos que necesitan los conectores. En particular, los archivos WSDL que describen los servicios configurados, los almacenes de certificados requeridos para invocarlos (keystores y truststores). Si la propiedad está comentada, tal como se ve en la Figura 2, los archivos se almacenarán por defecto bajo el directorio *data* de JBoss : “<conector-pge>/jboss-as/server/default/data/”
- **HOST:** Indica el nombre o dirección IP del equipo donde se instaló el Conector. Esta propiedad se utiliza cuando desde la consola de administración se entrega el link al WSDL que representa el servicio a

consumir para que sea usado por los clientes finales para invocar el servicio proxy. Al momento de instalar el conector deberá ingresar la IP del servidor donde lo instaló en este campo.

- **PORT\_TEST:** Puerto en donde el Conector escucha invocaciones a servicios publicados en el ambiente de testing de la PGE. En caso de no entrar en conflicto con ningún otro puerto, se recomienda dejar esta propiedad en su valor por defecto: 9800.
- **PORT\_PROD:** Puerto en donde el Conector escucha invocaciones a servicios publicados en el ambiente de producción de la PGE. En caso de no entrar en conflicto con ningún otro puerto, se recomienda dejar esta propiedad en su valor por defecto: 9700.
- **stsURL.production:** Url en donde se encuentra el servicio de STS del ambiente de producción. Tener en cuenta que esta ruta debe necesariamente hacer uso del nombre del servidor y no su dirección IP ya que en caso de especificar la ruta utilizando direcciones IP habrá un conflicto de certificados digitales. Al momento de escribir este tutorial, el valor de esta propiedad debe ser <https://servicios.pge.red.uy:10001/TrustServer/SecurityTokenServiceProtected>.
- **stsURL.test:** Url en donde se encuentra el servicio de STS del ambiente de testing. Tener en cuenta que esta ruta debe necesariamente hacer uso del nombre del servidor y no su dirección IP ya que en caso de especificar la ruta utilizando direcciones IP habrá un conflicto de certificados digitales. Al momento de escribir este tutorial, el valor de esta propiedad debe ser <https://testservicios.pge.red.uy:6051/TrustServer/SecurityTokenServiceProtected>.

## Ejecución

Luego de instalado y configurado el Conector como se explicó en la sección debe iniciarse el servidor JBoss AS. Para hacerlo, se debe ejecutar el script *run.sh* (en Linux) o *run.bat* (en MS Windows) que se encuentran en la carpeta `<conector-pge>/jboss-as/bin/`; en cualquier caso, 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 `<conector-pge>/jboss-as/bin/` y a continuación ejecutarlo directamente:

- **Linux:**
  - Opción 1
    - `cd <conector-pge>/jboss-as/bin/`
    - `./run.sh -b 0.0.0.0`
  - Opción 2
    - `./<conector-pge>/jboss-as/bin/run.sh -b 0.0.0.0`
- **Windows:**

- Opción 1
  - cd <conector-pge>\jboss-as\bin\
  - run.bat -b 0.0.0.0
- Opción 2
  - <conector-pge>\jboss-as\bin\run.bat -b 0.0.0.0

**Nota:** En algunas ocasiones, dependiendo de la configuración de red de la máquina, el conector puede demorar mucho en iniciar y, en ocasiones, puede fallar. Si este es el caso en su máquina, intente ejecutar el comando run el parámetro “-b 0.0.0.0”

**Importante:** Si inicia el conector de esta manera tenga en cuenta que sólo tendrá acceso local al servidor, esto es, que deberá acceder por la url *localhost.y* no por la ip de la máquina.

Cuando el servidor es ejecutado de esta manera, puede ser detenido con la combinación de teclas Ctrl-C en la consola donde está siendo ejecutado.

Una vez iniciado el servidor como fue explicado anteriormente, deberá verse en la consola la información de arranque y ejecución (log) del servidor similar a la mostrada en la Figura 3, donde se resalta que los componentes relativos al conector están instalados, el servidor está en ejecución, y no se visualizan excepciones de ningún tipo.

```

create esb service, agesic-sts.esb
create esb service, connector-pge-esb.esb
oyed: file:/C:/desarrollo/conector-pge_v4/jboss-as/server/default/deploy/agesic/conn
create esb service, esbUtils.esb
create esb service, jbossesb.esb
e/queue/DeadMessageQueue1 started, fullSize=200000, pageSize=2000, downCacheSize=20
e/queue/DataCollectorQueue1 started, fullSize=200000, pageSize=2000, downCacheSize=
e/queue/OperationsCollectorQueue1 started, fullSize=200000, pageSize=2000, downCach
e/queue/OperationsResultCollectorQueue1 started, fullSize=200000, pageSize=2000, do
indingService! Bound ConnectionManager 'jboss.jca:service=DataSourceBinding,name=JB
el java:/JBossESBDS datasource is already initialized
ger! initializing...
yl Using the non-legacy payload-to-message exchange pattern. To switch back to the
s.xml.
ing UDDI Client properties file file:/C:/desarrollo/conector-pge_v4/jboss-as/server/
istration @ servicebindingKeys
istration @ businessKeys
artz Scheduler v.1.5.2 created.
bStore initialized.
yl Quartz scheduler 'ESBScheduler:jbossesb.esb' initialized from an externally provi
yl Quartz scheduler version: 1.5.2
cheduler ESBScheduler:jbossesb.esb_$NON_CLUSTERED started.
create esb service, slsb.esb
create esb service, smooks.esb
create esb service, soap.esb
ation! Cannot read config.properties: Failed to locate resource '/config.properties'
ation! Cannot read config.properties: Failed to locate resource '/config.properties'
ation! Cannot read config.properties: Failed to locate resource '/config.properties'
ation! Cannot read config.properties: Failed to locate resource '/config.properties'
ation! Cannot read config.properties: Failed to locate resource '/config.properties'
ation! Cannot read config.properties: Failed to locate resource '/config.properties'
ation! Cannot read config.properties: Failed to locate resource '/config.properties'
ng org.apache.coyote.http11.Http11Protocol for http (coyote) invoker protocol handle
RemotingContentType value should be a String: null
icializando Coyote HTTP/1.1 en puerto http-192.168.10.202-9800
rancando Coyote HTTP/1.1 en puerto http-192.168.10.202-9800
ayListener! JBoss Remoting Gateway listener 'listener' started.
ng org.apache.coyote.http11.Http11Protocol for http (coyote) invoker protocol handle
RemotingContentType value should be a String: null
icializando Coyote HTTP/1.1 en puerto http-192.168.10.202-9700
rancando Coyote HTTP/1.1 en puerto http-192.168.10.202-9700
ayListener! JBoss Remoting Gateway listener 'listener' started.
rancando Coyote HTTP/1.1 en puerto http-192.168.10.202-8080
microKernel) [4.2.3.GA (build: SUNTag=JBoss_4_2_3_GA date=200807181417)] Started in

```

Figura 3: Inicio de servidor JBoss conteniendo al Conector PGE

Para confirmar que el conector funciona correctamente se debe acceder a su consola de administración, a través de la URL <http://host:puerto/conector-pge/>. Accediendo a esta URL se deberá ver la interfaz gráfica de la consola de administración tal como se ilustra en la Figura 4. Por defecto el usuario y contraseña es “conector”.



Figura 4: Página inicial de la consola de administración del Conector PGE

### 3. Configuración de Conectores

En las siguientes secciones se muestra paso a paso la configuración de un proxy del servicio Timestamp, y se ejemplifica un consumo del mismo.

#### Acceso

Para acceder al Conector una vez instalado y configurado como se especificó en la sección , se debe acceder a la consola de administración que se presentó en la Figura 4. Se debe proporcionar un usuario y una contraseña; por defecto, y a menos que se modifique, debe especificarse “conector” tanto usuario como contraseña.

Al acceder a la consola de administración del conector se desplegará una pantalla similar a la que muestra la Figura 5, en la cual se aprecia el listado de conectores y una lista de selección que permite seleccionar el ambiente sobre el cual se desea trabajar, en este caso: testing.

1 Sustituir “host” por el nombre o dirección IP del equipo donde ejecuta el Conector, y “port” por el número de puerto por el que atiende el servidor JBoss AS (por defecto es 8080).

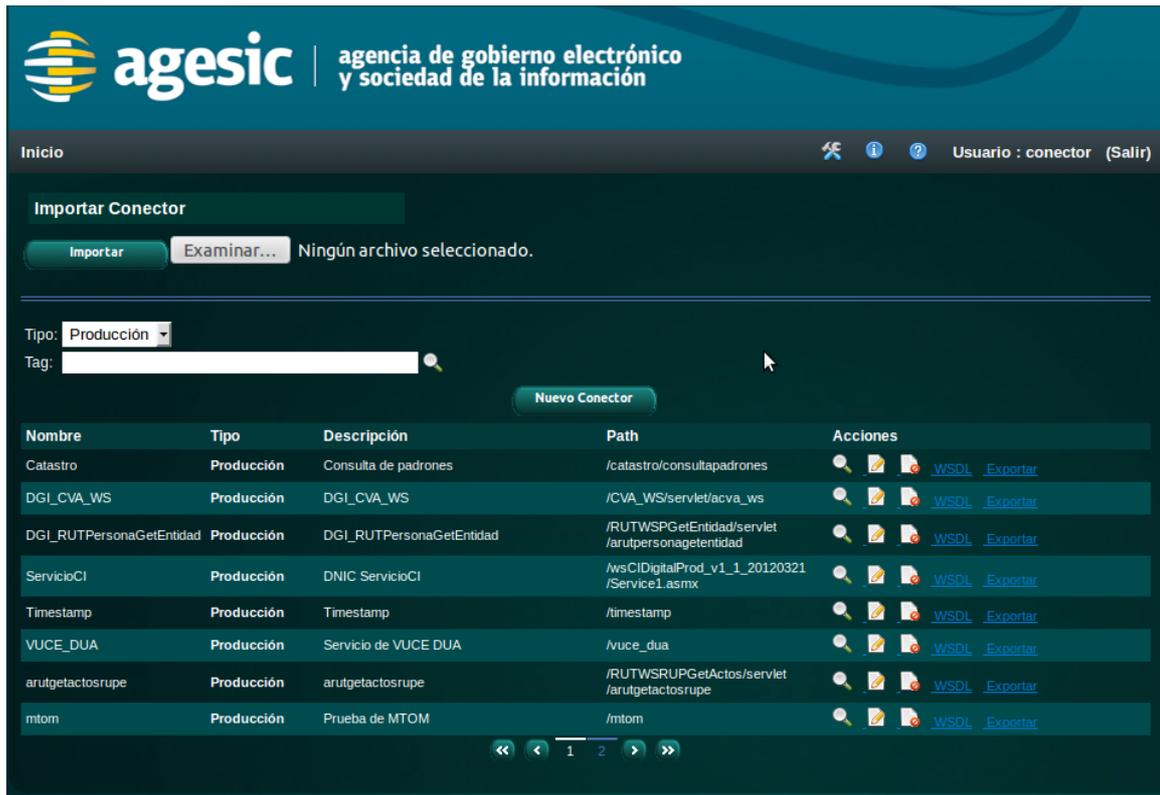


Figura 5: Listado de Conectores para ambiente de producción

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.

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 6 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.



Figura 6: Botones de acción para los conectores

## Crear un nuevo Conector

Para crear un conector se debe presionar el botón “**Nuevo Conector**” que aparece en la pantalla inicial.

Ingresar los datos para la totalidad de los campos solicitados, como se muestra en la figura Figura 7. Si se omite alguno de los campos requeridos, al momento de presionar el botón *Salvar* se volverá a mostrar la misma página indicando cual es el o los campos en los cuales no se ingresaron datos.

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.

**IMPORTANTE:** Para esta parte del tutorial deberá haber completado el “tutorial de certificados plataforma JAVA”.

Es necesario:

- Keystore con certificado SSL emitido por la CA de Agestic y la contraseña correspondiente. ***Este keystore se utilizará tanto para “Keystore SSL”, como para “Keystore de Organismo”.***
- Truststore conteniendo la clave pública del certificado que presenta la PDI.
- WSDL del servicio a consumir en este tutorial: Timestamp. El archivo puede ser descargado en el FTP de agestic:

<sftp.agesic.gub.uy/E-Learning/ConectorPGE/TimestampServicewsd1.wsdl>

Figura 7: Pantalla creación de un conector

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).

- **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 “Test” y “Producción”. En el caso del ejemplo se selecciona “Test” 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.

- **Wsa:To:** este dato será agregado por el Conector como encabezado (header) del mensaje SOAP que enviará a la PGE. Corresponde a un endpoint lógico, que permite a la PGE determinar cuál es el servicio concreto que se desea invocar. Para el caso del ejemplo, se colocará el siguiente valor “*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 y debe ser proporcionado por el organismo consumidor.
- **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*”.
- **Tipo de Token:** indica el tipo de token que debe generarse para permitir la invocación de servicios. En el caso del ambiente de testing, el valor de este campo debe ser *urn:tokensimple*. En este ejemplo, como se desea consumir un servicio del ambiente de testing, se utilizará el valor “*urn:tokensimple*”.

Deberá habilitar el checkbox “*Habilitar configuración local*” ya que aparecerá deshabilitado por defecto.

- **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. *Deberá poner el alias del keystore creado en el tutorial de certificados JAVA.*
- **Password del Keystore de Organismo:** este campo contiene la contraseña del almacén de claves especificado por el campo *Keystore de Organismo*. *Deberá poner el password del keystore creado en el tutorial de certificados JAVA*
- **Password Keystore SSL:** este campo contiene la contraseña del almacén de claves especificado por el campo *Keystore de SSL*. *Deberá poner el alias del keystore SSL creado en el tutorial de certificados JAVA (el mismo que el anterior, dado que se utilizará el mismo keystore para ambos campos).*
- **Password Truststore:** este campo contiene la contraseña del almacén de claves especificado por el campo *Truststore de SSL*. *Deberá poner el password del truststore SSL creado en el tutorial de certificados JAVA.*
- **WSDL:** este campo permite ingresar la ruta al archivo WSDL que describe el servicio que se desea consumir. En el ejemplo, se carga el archivo *TimestampServicewSDL.wSDL* descargado del FTP de Agesic.
- **Keystore de Organismo:** este campo permite ingresar la ruta al keystore de organismo utilizado para la firma del mensaje SOAP. *Deberá poner el keystore creado en el tutorial de certificados JAVA.*
- **Keystore de SSL:** este campo permite ingresar la ruta al keystore utilizado para la comunicación HTTPS. *Deberá poner el keystore SSL*

creado en el tutorial de certificados JAVA (el mismo que el anterior, dado que se utilizará el mismo keystore para ambos campos).

- **Truststore de SSL:** este campo permite ingresar la ruta al truststore utilizado para la comunicación HTTPS con autenticación mutua. *Deberá poner el truststore creado en el tutorial de certificados JAVA.*

La figura 8 muestra los campos ya cargados con los datos apropiados; para continuar con la creación del conector, se debe presionar el botón refrescar que está a la derecha del campo WSDL. Este botón (resaltado en rojo), permite cargar los archivos seleccionados, y a su vez, determinar las operaciones especificadas en el WSDL. Por lo tanto, una vez presionado dicho botón, se cargará automáticamente la sección *Rol – Operación* de la parte inferior con una tabla para que se especifique manualmente los campos *Rol* y *wsa:Action* que deben ser utilizados cuando sea invocada cada operación.

The screenshot shows a web-based configuration interface for a connector. The top bar includes a user profile 'Usuario : conector (Salir)'. The main form contains the following fields and options:

- Nombre:** Timestamp
- Tipo:** Test
- Descripción\*:** Timestamp
- Path\*:** /timestamp
- Url:** https://testservicios.pge.red.uy:6055/timestamp/TimestampService
- wsa:To\*:** http://testservicios.pge.red.uy/timestamp\_reportes
- Username\*:** mcaponi
- Organismo\*:** Agesic
- Tipo de Token\*:** urn:tokensimple
- Ingresar credenciales de Username Token
- Habilitar Configuración Local
- Habilitar cache de Tokens
- Alias del Keystore Organismo\*:** 0f026f823ca3597ced3953188b1628de\_be45dff3-4f56-4728-8332-77080b0c1
- Password Keystore Organismo\*:** [Redacted]
- Password Keystore SSL\*:** [Redacted]
- Password Truststore\*:** [Redacted]
- Keystore Organismo\*:** Examinar... Ningún archivo seleccionado.
- Keystore SSL\*:** Examinar... Ningún archivo seleccionado.
- Truststore SSL\*:** Examinar... Ningún archivo seleccionado.
- WSDL\*:** Examinar... Ningún archivo seleccionado. (Refresh button highlighted in red)
- Tag:** [Empty field]

**Rol - Operación**

Rol	wsa:Action	Operación
ou=gerencia de proyectos.c	[Empty field]	GetTimestamp

Buttons at the bottom: Cancelar, Salvar, Borrar, Pasar a Producción

Figura 8: Pantalla de conector con datos cargados

**Nota:** en caso de no completarse el campo *wsa:Action* este deberá completarse de la siguiente manera:

- Testing: <http://testservicios.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.

Rol	wsa:Action	Operación
OU=TEST_TUTORIAL	http://test.pge.red	GetTimestamp

Figura 9: Sección Rol-Operación habilitada luego de que fueron parseadas las operaciones en el WSDL

El resultado luego de presionar el botón mencionado puede apreciarse en la figura 9 (en este caso el servicio tiene una sola operación, por lo que la tabla tendrá una sola fila; siempre la tabla contendrá una fila para cada operación descrita en el WSDL).

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://testservicios.pge.red.uy/timestamp (confirmarlo observando el WSDL).*

Una vez finalizada la carga, se presiona el botón *Salvar*, y se ha terminado de crear el conector. Luego de esta operación se observará una pantalla similar a la de la 15, 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 como realizar operaciones de mantenimiento sobre este conector.

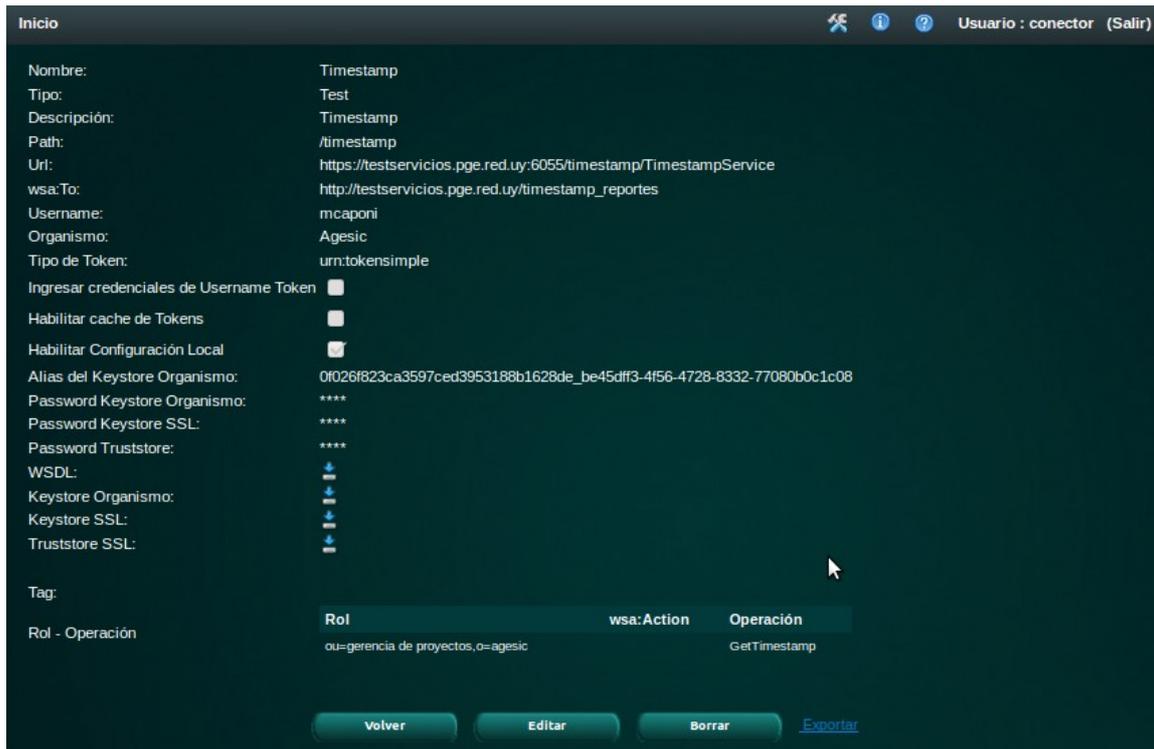


Figura 10: Pantalla que se despliega luego de dar de alta un conector

## Consumir un servicio en el Conector:

Utilizaremos el software SoapUI que es para pruebas funcionales de servicios SOAP.

1. Seleccionamos el wsdl del servicio que queremos consumir. En este caso, vamos a utilizar el servicio Timestamp, que configuramos anteriormente en el conector.

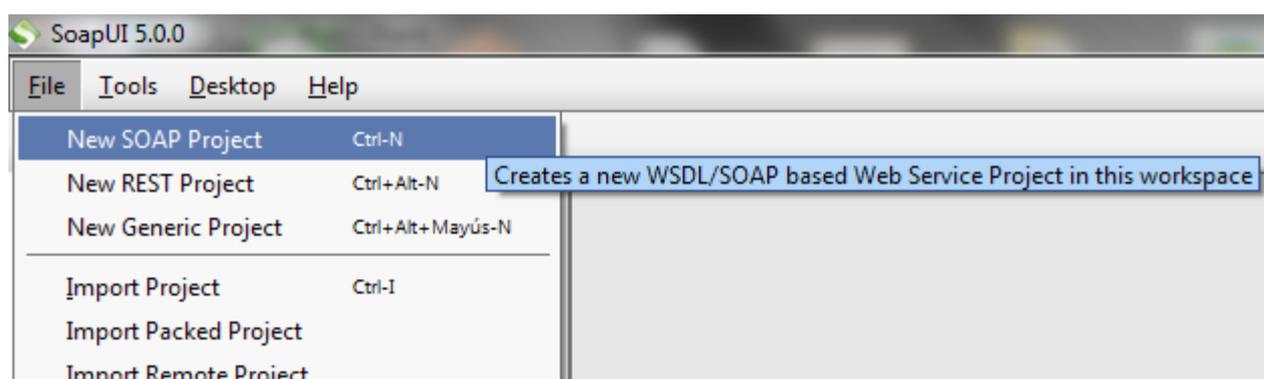


En nuestro navegador se abrirá una nueva pestaña, mostrándonos el archivo wsdl que define al servicio:

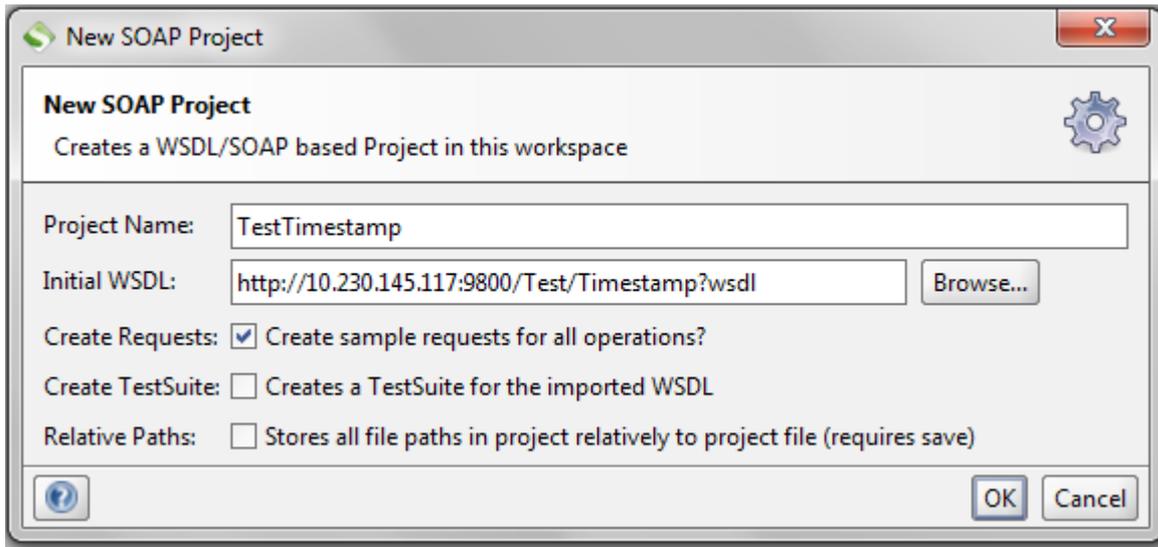


2. Copiamos la URL de modo que nos quede en el portapapeles.

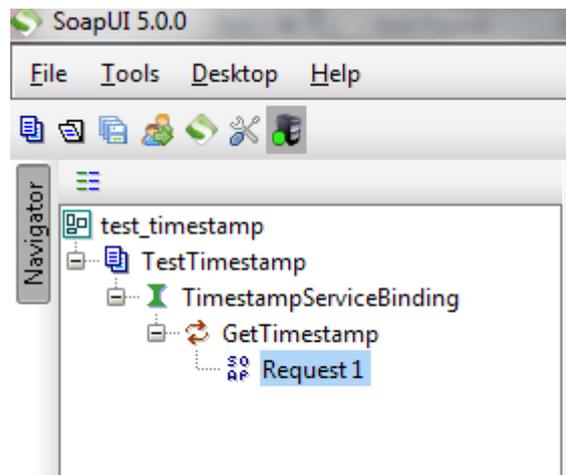
3. Creamos un nuevo proyecto SOAP en el SoapUI:



4. Pegamos la URL en donde dice "Initial WSDL" e ingresamos un nombre al proyecto:



5. SoapUI generará un pedido de ejemplo para el servicio. Lo seleccionamos:



6. Hacemos click en el botón “play”. En la segunda sección de la ventana se mostrará la respuesta.

