

## WebServices SOAP en sistema de Certificados de Origen

### INTRODUCCIÓN

SOAP (Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es un paradigma de mensajería de una dirección sin estado. Por ejemplo, un mensaje SOAP que podría ser enviado a un web service para realizar la búsqueda de algún dato en una base de datos, indicando para ello los parámetros necesitados en la consulta. El servicio podría retornar un documento en formato XML con el resultado.

En un escenario básico, los nodos SOAP se comunican uno asumiendo el rol de SOAP Sender y otro el de SOAP Receiver.

WSDL (Web Services Description Language), es un formato del Extensible Markup Language (XML) que se utiliza para describir servicios web.

WSDL describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

El WSDL nos permite tener una descripción de un servicio web. Especifica la interfaz abstracta a través de la cual un cliente puede acceder al servicio y los detalles de cómo se debe utilizar.

WSDL se usa a menudo en combinación con SOAP y XML Schema. Un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar qué funciones están disponibles en el servidor. Los tipos de datos especiales se incluyen en el archivo WSDL en forma de XML Schema. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL.

El WSDL nos permite tener una descripción de un servicio web. Especifica la interfaz abstracta a través de la cual un cliente puede acceder al servicio y los detalles de cómo se debe utilizar.

Así trabaja SOAP.

En las Entidades asociadas que utilizan nuestras aplicaciones informáticas para la gestión y emisión de Certificados de Origen, implementamos SOAP 1.1 y WSDL 2.0 en un computador, que con sus aplicaciones asume el rol de SOAP Sender (publicadora de web services). A partir de esto, terceras partes (exportadores, despachantes o sedes/delegaciones) podrán desarrollar sus propias aplicaciones SOAP Receiver (consumidoras de web services), con el objetivo de integrar sistemas y automatizar operaciones usando estándares del mercado.

Los web services SOAP reciben y envían información en documentos XML. Para el caso particular de operar con documentos binarios (XML firmados digitalmente, PDF, ZIP o similares), usaremos el estándar de codificación Base64. Este estándar lo que hace es codificar bytes en caracteres ASCII imprimibles (pasa un archivo binario a un archivo equivalente en ASCII). Actualmente muchos lenguajes de programación incluyen este estándar. También hay numeras aplicaciones opensource que permiten codificar/decodificar binarios desde/hacia Base64. Datos de configuración que publique el servidor, como por ejemplos tablas y/o datasets con parámetros o respuestas a peticiones, serán enviados a través de un campo de caracteres (string) que contenga un archivo XML codificado en UTF-8.

### PRE REQUISITOS

El usuario que consuma los webservices SOAP debe estar registrado en la base de datos de la Entidad como Exportador, Despachante de Aduana o Sede/Delegación autorizada. Además del registro, debe tener acceso a la aplicación de Certificados de Origen en el Website de la Entidad, usando un nombre de Usuario/Contraseña asignados. Debe poder operar con esta aplicación web sin dificultad (la condición ideal es que opere habitualmente sus Certificados de Origen desde el website de la Entidad).

El software del usuario que consuma los web services SOAP o la aplicación del usuario vinculado a los mismos, debe tener la capacidad de generar un archivo XML que contenga los datos del Certificado de Origen conforme a la normativa y versión vigentes de ALADI ([https://drive.google.com/file/d/1qkYk0U6be2KE8-GGGLrMES26kQ8XK\\_dS/view?usp=sharing](https://drive.google.com/file/d/1qkYk0U6be2KE8-GGGLrMES26kQ8XK_dS/view?usp=sharing)). Se podrán enviar documentos XML con formato ALADI para todos los acuerdos comerciales, sin importar si el país de destino admite COD o no. En otras palabras, los webservices SOAP aplican a certificados de origen en papel y digitales. Para el caso de que no pueda generar un archivo XML con este formato, podrá usar un webservice SOAP de conversión destinado a recibir un archivo CSV delimitado por punto/ coma y transformarlo en XML de ALADI. En tal caso, este archivo CSV se debe ajustar a las especificaciones técnicas establecidas oportunamente (consultar documentación técnica al respecto).

La documentación enviada a través de los webservices SOAP (Certificado de Origen en formato XML de ALADI, más facturas de exportación y declaración jurada) deberán cargarse firmados digitalmente con un certificado digital (CID) que pueda la Entidad validar en su propio registro de firmas. En caso del documento XML deberá firmarse con la aplicación DS-COD existente para tal fin.

## IMPLEMENTACIÓN

El servidor de producción de la Entidad tiene habilitado el web server que opera en el puerto 80. Este es el puerto por defecto que utiliza el website. Los web services SOAP operarán en el puerto 8080 (o en el puerto que se indique en cada caso). La implementación de los webservices SOAP está hecha con el software Apache Tomcat 8.

Para realizar pruebas, la comunicación entre el SOAP Sender (servidor) y el SOAP Receiver (cliente) no será encriptada. Luego la cifraremos con la licencia de los Certificados Digitales adecuados a tal fin.

El WSDL de los WebServices SOAP publicado en el servidor de producción de las entidades tendrán este formato de URL:

<http://subdominio.dominio.com.ar:8080/wsa/wsa1/wsd1?targetURI=urn:localhost-org>

- el subdominio.dominio.com.ar variará en función de los nombres que defina la Entidad
- el puerto 8080 puede variar según indicación que se realice para cada caso
- se puede ver directamente en XML o bien con formato instalando alguna extensión en el navegador

La documentación adicional que genera el sistema con ejemplos de uso de la interface (en XML y 4GL/ABL de Progress) estará en una URL similar:

[http://subdominio.dominio.com.ar/sauken/doc/service\\_urn\\_localhost\\_org\\_WSSaukenService.html](http://subdominio.dominio.com.ar/sauken/doc/service_urn_localhost_org_WSSaukenService.html)

- el subdominio.dominio.com.ar variará en función de los nombres que defina la Entidad

El test de cada uno de los webservices SOAP se puede realizar con la aplicación opensource SOAP-UI. La misma se puede descargar para su versión Windows de 64 bits desde esta URL:

[https://drive.google.com/file/d/1-uph36jW0PlhP0eCPCpHsc\\_O3F2e7tfu/view?usp=sharing](https://drive.google.com/file/d/1-uph36jW0PlhP0eCPCpHsc_O3F2e7tfu/view?usp=sharing)

Dentro de SOAP-UI se debe seleccionar webservices SOAP y poner esta dirección de WSDL:

<http://subdominio.dominio.com.ar:8080/wsa/wsa1/wsd1?targetURI=urn:localhost-org>

- el puerto 8080 puede variar según indicación que se realice para cada caso

Luego crear los Test Case para finalmente simular todo el trabajo a mano (Login, API y Logout). Es muy sencillo de usar. Todas las respuestas están en XML. En **CODLogin** acceder con el usuario y contraseña que se utiliza habitualmente para acceder a la aplicación del website. Cuando se utilicen funciones de la API se debe copiar los datos devueltos por **CODLogin** como entrada de los mismos (**WebID**, **SessionID**). Las sesiones deben concluirse con **CODLogout**.

En **ANEXO-1** hay una breve explicación de cómo utilizar SOAP-UI para realizar las pruebas.

En **ANEXO-2** están enumerados todos los webservices SOAP disponibles.

Los principales web services SOAP a utilizar son:

### A. Autenticación

La autenticación es en base a informar un usuario y contraseña. El sistema usa los mismos usuarios/contraseñas de acceso del website. Existen 4 tipos de usuarios: exportadores, despachantes de aduanas, delegaciones y usuario interno. Nos remitiremos siempre al "usuario exportador" para simplificar el proceso de Login (todos los otros requieren de otro webservice para la selección de un exportador con el cual se operará luego de Login).

El webservice "**CODLogin**" recibe de entrada usuario/contraseña y si la respuesta es exitosa, devuelve un WebID (un número entero creciente) y un SessionID (es un GUID -cadena de caracteres- usado como token).

El sistema en su base de datos asocia el usuario/contraseña ingresado (al exportador que va a operar) con el WebID y el SessionID, registrando a partir de esto los datos de la sesión (exportador por ejemplo) y todas las operaciones con registro que se realicen.

La SessionID (el token) caducará cuando se realice Logout o bien con cada cambio de día (a las 0:00 hs de cada nuevo día caducan los token del día anterior).

El webservice "**CODLogout**" caducará el SessionID (token) y terminará exitosamente el WebID.

Todas las operaciones que se realicen con otros webservices deberán ejecutarse dentro del marco de los "**CODLogin**" y "**CODLogout**". Todos los webservices ejecutados dentro de este marco referirán de manera obligatoria en cada una de sus llamadas los parámetros WebID y SessionID.

En el caso de que **"CODLogin"** acceda un usuario tipo DESPACHANTE o DELEGACIÓN será necesario para continuar interactuando seleccionar un Exportador con el cual operar con el webservice **"CODSeleccionaExpo"** Este Exportador seleccionado persistirá hasta tanto se seleccione otro o bien se ejecute **"CODLogout"** La lista de Exportadores para el tipo de usuario conectado se puede obtener con el webservice **"CODExpoVinculados"** Para asegurarse un **"CODSeleccionaExpo"** exitoso, verifique que el CUIT del Exportador con el cual desea operar esté dentro de la lista entregada por **"CODExpoVinculados"**

Se debe verificar siempre el estado de conexión y del Exportador seleccionado con el webservice **"CODEstado"** Este webservice debe devolver en **"estado"** un **"yes"** ("true" es igual) para asegurarse que estamos conectados (en login). Luego, si eso sucede, en **"expo-asignado"** debe devolver otro **"yes"**. En caso de un **"no"** ("false" es igual), implica que no hay un Exportador seleccionado y debe ejecutarse **"CODSeleccionaExpo"** para seleccionar uno válido. Si **"CODSeleccionaExpo"** es exitoso, entonces **"expo-asignado"** cambiará a **"yes"** y en **"cdeoCOMP"**, **"cuitCOMP"**, **"cdeoNomexp"** aparecerán los datos del Exportador seleccionado.

## B. Tablas de operación

Los webservices entregan las principales tablas de parametrización del sistema. Tablas que se deberán actualizar con cierta periodicidad (una vez por mes o cada trimestre) debido a que las mismas pueden sufrir cambios.

Las tablas de parametrización serán entregadas en formato XML codificadas en "UTF-8 Tablas anidadas (una tabla con muchas relaciones en otra y en otra) serán puestas como "datasets" dentro del XML (ejemplo la salida de **"CODTablaNormasOrigen"**).

Los principales webservices SOAP de parametrización son:

<b>"CODTablaPaíses"</b>	Devuelve la tabla ISO-3166 de países
<b>"CODTablaProvincias"</b>	Devuelve la tabla de provincias argentinas
<b>"CODTablaUniMedida"</b>	Devuelve la tabla con unidades de medidas
<b>"CODTablaTransportes"</b>	Devuelve la tabla con transportes
<b>"CODTablaAcuerdos"</b>	Devuelve la tabla con Acuerdos Comerciales
<b>"CODTablaNormasOrigen"</b>	Devuelve el dataset con Acuerdos Comerciales y sus Normas de Origen asociadas
<b>"CODTablaNCM"</b>	Devuelve la tabla con Nomenclador Común del Mercosur NCM
<b>"CODTablaNALADI"</b>	Devuelve la tabla con Nomenclador NALADI
<b>"CODTablaNALADISA"</b>	Devuelve la tabla con Nomenclador NALADI-SA

No es necesario tomar las tablas con los nomencladores. Queda a criterio del implementador utilizarlas o no.

## C. Transacciones

Cada webservice en particular involucra a una transacción simple. Al convocar un webservice se inicia una transacción en la base de datos. Si el resultado de la ejecución del mismo es exitosa, la transacción se aplica (commit). En caso de resultar negativa, la transacción de vuelve atrás (rollback).

Existen procesos que involucran la ejecución de más de un webservice para completarlo. En ese caso, hay que aplicar dos webservices específicos: **"CODBeginTransaction"** el cual iniciará una transacción asíncrona y devolverá un **TrasactionID**, y el **"CODEndTransaction"** que terminará la misma. Un **"CODRollbackTransaction"** deshacerá los cambios realizados, o bien los mismos volverán atrás cuando caduque el **SesionID** (cambio de día), o cuando se inicie una nueva **"CODBeginTransaction"** Estos procesos son muy puntuales y poco usados. Normalmente se utilizan transacciones simples dentro del mismo webservice invocado. Ejemplo de uso: cuando envío documentación complementaria de un Certificado de Origen: inicio una transacción, tomo un Certificado de Origen, envío el documento XML firmado, luego envío la factura de exportación, y luego la declaración jurada. Al terminar termino esa transacción.

Si se está dentro de un **"CODBeginTransaction"** y quiere volver atrás, tome el **TrasactionID** y deshaga todo con el **"CODRollbackTransaction"**. Si pierde el **TrasactionID**, deberá realizar un **"CODLogout"** para poder seguir operando. Esa transacción abierta queda en un estado que llamamos "limbo" y se volverá atrás de manera automática al finalizar el día. El usuario no verá ni podrá operar con transacciones abiertas.

#### D. Webservices API

Uno de los webservices principales es el que informa el estado de operación del usuario en Login. Es importante utilizarlo para conocer el mismo. El webservice se llama "CODEstado" y entrega esta salida:

```
CODEstado
result ?
Estado yes

Usuario C44218@P34946
Delegación CC

Compañía 45089
CUIT Exportador 30707228225
Razón Social Compo Expert Argentina S.R.L.

Persona 34946
Nombre Patricio Novaro

Tipo Otro 2

Exportador Otro 44218
CUIT Otro 30709006629
Razón Social Otro Ag Comercio Exterior

Apto COD yes

Tipos Habilitados
ALADI,COMUN,MERC/BOLIVIA,MERC/CHILE,MERC/CO-ECU-VEN,MERC/C
OLOMBIA,MERC/INDIA,MERC/ISRAEL,MERCOSUR

Entidad CAC

Tipo Login DESPACHANTE
Exportador Asignado yes

Mensajes Usuario: Patricio Novaro - Despachante: Ag Comercio Exterior
- Exportador: Compo Expert Argentina S.R.L. [CUIT: 30707228225]
(Habilitado COD)
```

En esta salida hay datos importantes para tener en cuenta: con que exportador estoy operando, tipos de certificados de origen habilitados (en varios webservices el tipo de certificado es necesarios que se pase como parámetro), si el exportador está en condiciones de emitir COD y similares.

Enumeramos y describimos algunos de los webservices API. En general por cada caso de uso implementado en el website existe un webservice que lo representa.

#### "CODVerifica"

Valida la existencia de un Certificado de Origen y devuelve si se trata de COD o Papel, más el estado del mismo (estados posibles: 0 = Error, 1 = No Existe, 2 = Anulado, 3 = En carga web, 4 = En proceso en entidad, 5 = Enviado, 6 = Aprobado, 7 = Rechazado, 8 = Emitido Papel, 9 = Emitido COD, 10 = Publicado para descarga web y pagado), 11/12/13 iguales a 8/9/10 pero que fueron cargados manualmente por la Entidad.

#### "CODValida"

Permite validar los datos de un Certificado de Origen. Devuelve errores en caso de existencia de los mismos.

#### "CODDichoPartida"

Devuelve el dicho de partida de una posición arancelaria y el formato de visualización correcto de la misma. El implementador puede optar por tomar el dicho de partida de la posición arancelaria usando este webservice o bien puede usar su propia leyenda a la hora confeccionar el Certificado de Origen.

#### "CODFirmasAutorizadas"

Devuelve una tabla con las firmas autorizadas y vigentes del Exportador en operación. Es importante asignar un nombre con la Firma Autorizada del Exportador al Certificado de Origen que esté en esta lista. Si el Certificado de Origen es Digital (COD) es relevante que el nombre seleccionado esté habilitado para Operar con COD. *En la última actualización agregamos el número de serie del certificado digital registrado en la base de datos de la Entidad.*

**Pre-carga de Certificado de Origen** (estos webservices son equivalentes a la carga manual o a la carga a través de archivo delimitado CSV disponible en el website, no se usan para enviar documentación electrónica conjunta y firmada digitalmente)

#### "CODUploadCSV"

Permite cargar un archivo delimitado de texto CSV que contiene datos de un Certificado de Origen. Este webservice realiza las mismas operaciones disponibles en el website: "Enviar Archivo de Carga/Archivo Delimitado CSV", y luego "Enviar Carga a la Entidad La información cargada con este webservice podrá ser editada desde el website y le aparecerá al operador del backend como si hubiera sido enviada desde el website (se le informa que el origen es un webservice). A partir de acá el funcionario del backend operará de modo normal con el mismo.

#### "CODDownloadPDFPre"

Permite la descarga de un documento PDF para la pre-visualización del Certificado de Origen que se acaba de cargar a través de un archivo delimitado CSV. Este documento PDF saldrá con la leyenda "No Válido / No Legal" ya que sólo se genera a los fines de mostrar en un formato amigable los datos que acaban de cargarse.

#### "CODUploadFactura" y "CODUploadDDJJ"

Al utilizar "CODUploadCSV" es posible adjuntar luego la factura de exportación y la declaración jurada. A diferencia del website, acá se adjunta el documento luego de ser enviado a la entidad (no antes). Sólo se permite adjuntar documentación al Certificado de Origen mientras esté pendiente de Aprobación o Rechazo. Nunca se operará con Certificados de Origen Emitidos o Anulados.

#### "CODDownloadXMLExportador"

Permite la descarga del documento XML generado (normalmente a partir del "CODUploadCSV") para que el exportador aplique luego su firma digital sobre el mismo.

#### "CODUploadXMLExportador"

Permite la carga a la base de datos el documento XML firmado digitalmente por el exportador.

**Carga de Certificados de Origen a través de documento XML** (estos webservices permiten cargar directamente al sistema un Certificado de Origen en formato de documento XML con o sin firma digital del Exportador y sus documentos vinculados: facturas de exportación y declaración jurada, estos webservices son similares a los disponibles en el website del sistema)

#### "CODEXPUploadXML"

Permite la carga a la base de datos un documento XML con un Certificado de Origen. Este documento XML puede estar firmado digitalmente o no. En caso de acuerdos que operen con COD deben enviarse firmados digitalmente de manera obligatoria.

### "CODEXPUploadFactura" y "CODEXPUploadDDJJ"

Los documentos XML enviados con "CODEXPUploadXML" deben tener obligatoriamente vinculados la/s factura/s de exportación y la declaración jurada. Estos webservices deben ser utilizados para realizar esta tarea. *El parámetro de entrada "filetype" se utiliza para informar el tipo de contenido del archivo del archivo binario codificado en "base64" que se envía. Generalmente el "filetype" contendrá un "pdf" o bien un "zip" por ejemplo.*

### "CODDownLoadPDFImpresion"

Permite la descarga del PDF de impresión para la pre-visualización (si se tratara de un COD) o bien para la impresión definitiva en los formularios entregados a tal fin (en el caso de un Certificado de Origen en Papel).

### "CODDownLoadXMLEmitido"

Permite la descarga del documento XML que contiene un COD definitivo. *El parámetro de entrada "Zip" podrá contener un "yes" o "true", o bien un "no" o "false". Se lo utiliza para indicar si desea descargar el archivo XML con el COD comprimido con la aplicación ZIP o no. En caso de usar "comprimido" siempre el binario resultante de archivo codificado "base64" tendrá una extensión ".zip" (se la deberá poner el implementador del programa que consume el webservice SOAP). Caso contrario se deberá asignar ".xml". Recomendamos utilizar siempre formato comprimido, ya que dentro del mismo el documento XML tendrá el nombre adecuado y conforme a las normativas de ALADI. En caso de descargar el XML y requerir el nombre a imponer, puede usar "CODVerifica" y tomar el mismo de "Certificateld".*

#### Ejemplo de uso:

El proceso completo de envío a la entidad de un documento XML firmado digitalmente por el exportador, junto con la documentación complementaria (factura de exportación y declaración jurada) requiere realizar las siguientes acciones:

"CODLogin"  
"CODBeginTransaction"  
"CODEXPUploadXML"  
"CODEXPUploadFactura"  
"CODEXPUploadDDJJ"  
"CODEndTransaction"  
"CODLogout"

*Notar que al iniciar "CODBeginTransaction" correctamente devuelve un "TransactionID". Este dato será necesario aplicar a los procesos subsiguientes. Al terminar correctamente "CODEndTransaction" devuelve un "Valor" con el cual podrá referirse en procesos de control para verificar el efectivo ingreso de los datos al sistema de la Entidad (se lo usa también para realizar seguimiento de las acciones realizadas). Una vez que la información enviada haya sido aceptada por la Entidad, se utilizará el "codigo" de trámite de Certificado de origen asignado para todas las operaciones siguientes.*

El seguimiento de la documentación enviada, como la aceptación o rechazo de la misma se la puede consultar y obtener a partir de los webservices "CODTablaCarga" y "CODdatoCarga".

A continuación un extracto de un ejemplo real de uso de estos webservices para cargar los documentos en el sistema (escrito en 4GL/ABL de Progress):

```
.....  
if search(f_XML) = ? or search(f_FACT) = ? or search(f_DDJJ) = ? then /* URLs en almacenamiento de los archivos a cargar */  
do:  
    message "No existen los archivos a cargar !!" view-as alert-box error.  
    return.  
end.  
  
copy-lob from file f_XML to compbase64 no-convert.  
XMLbase64 = base64-encode(compbase64). /* Longchar con archivo a cargar en Base64 */  
  
copy-lob from file f_FACT to compbase64 no-convert.  
BLOBbase64-FACT = base64-encode(compbase64). /* Longchar con archivo a cargar en Base64 */  
  
copy-lob from file f_DDJJ to compbase64 no-convert.  
BLOBbase64-DDJJ = base64-encode(compbase64). /* Longchar con archivo a cargar en Base64 */  
  
/*-----*/  
run CODLogin in hWSSaukenObj (input xUsuario, input xContrasenia, output result, output validado, output xWebID, output xSessionID,  
    output cMensaje, output cEstado).  
message "Usuario:" xUsuario skip "Contrasena" xContrasenia skip "WebID" xWebID skip "result" result skip "Validado" validado skip  
    "SessionID" xSessionID skip "Mensaje" cMensaje skip "Estado" cEstado view-as alert-box title "CODLogin".  
if not validado then  
    return.  
/*-----*/  
  
run CODEstado in hWSSaukenObj (input xWebID, input xSessionID, output result, output cdeo-estado, output cdeo-usr, output cdeo-deleg,  
    output cdeo-comp, output cdeo-pers, output cdeo-nomexp, output cdeo-apenom, output cdeo-tipotr,
```

```

        output cdeo-codotr, output cdeo-nomotr, output cuit-comp, output cuit-otr, output apto-cod,
        output lista-tipos, output entidad, output tipo-login, output exp-asignado, output cdeo-mensaje).
message "result" result skip "Estado" cdeo-estado skip(1) "Usuario" cdeo-usr skip "Delegación" cdeo-deleg skip(1)
"Compañía" cdeo-comp skip "CUIT Exportador" cuit-comp skip "Razón Social" cdeo-nomexp skip(1)
"Persona" cdeo-pers skip "Nombre" cdeo-apenom skip(1)
"Tipo Otro" cdeo-tipotr skip(1)
"Exportador Otro" cdeo-codotr skip "CUIT Otro" cuit-otr skip "Razón Social Otro" cdeo-nomotr skip(1)
"Apto COD" apto-cod skip(1) "Tipos Habilitados" lista-tipos skip(1) "Entidad" entidad skip(1)
"Tipo Login" tipo-login skip "Exportador Asignado" exp-asignado skip(1)
"Mensajes" cdeo-mensaje view-as alert-box title "CODEstado".

/*-----*/

run CODBeginTransaction in hWSSaukenObj (input xWebID, input xSessionID, output result, output xTransID, output procesado,
output cEstado).
message "TransID:" xTransID skip "result" result skip "Procesado" procesado skip "Estado" cEstado
view-as alert-box title "CODBeginTransaction".
if not procesado then
do:
message "Termina: CODBeginTransaction".
return.
end.

/*-----*/

run CODEXPUploadXML in hWSSaukenObj (input xWebID, input xSessionID, input xTransID, input XMLbase64, output result, output procesado,
output cEstado).
message "result" result skip "Procesado" procesado skip "Estado" cEstado view-as alert-box title "CODEXP-UploadXML".
if not procesado then
do:
message "Termina: CODEXPUploadXML".
return.
end.

/*-----*/

run CODEXPUploadFactura in hWSSaukenObj (input xWebID, input xSessionID, input xTransID, input "pdf", input BLOBbase64-FACT,
output result, output procesado, output cEstado).
message "result" result skip "Procesado" procesado skip "Estado" cEstado view-as alert-box title "CODEXP-UploadFactura".
if not procesado then
do:
message "Termina: CODEXPUploadFactura".
return.
end.

/*-----*/

run CODEXPUploadDDJJ in hWSSaukenObj (input xWebID, input xSessionID, input xTransID, input "pdf", input BLOBbase64-DDJJ,
output result, output procesado, output cEstado).
message "result" result skip "Procesado" procesado skip "Estado" cEstado view-as alert-box title "CODEXP-UploadDDJJ".
if not procesado then
do:
message "Termina: CODEXPUploadDDJJ".
return.
end.

/*-----*/

run CODEndTransaction in hWSSaukenObj (input xWebID, input xSessionID, input xTransID, output result, output procesado, output cEstado,
output webcargaID).
message "TransID:" xTransID skip "result" result skip "Procesado" procesado skip "Estado" cEstado skip "WebCargaID" webcargaID view-as
alert-box title "CODEndTransaction".
if not procesado then
do:
message "Termina: CODEndTransaction".
return.
end.

/*-----*/

run CODDatoCarga in hWSSaukenObj (input xWebID, input xSessionID, input integer(webcargaID), output result, output lc_carga).
message "result" result view-as alert-box title "CODDatoCarga".

estado = hds_carga:read-xml("longchar",lc_carga,?,?).

delete object hds_carga.

for each t_carga:
display t_carga with frame carga 1 down 1 columns overlay centered width 160 view-as dialog-box
title "Registro de Carga Web (web_carga)".
end.

hide frame carga.

/*-----*/

run CODLogout in hWSSaukenObj (input xWebID, input xSessionID, output result, output zSessionID, output cEstado).
message "result" result skip "Anterior SessionID" zSessionID skip "Estado" cEstado view-as alert-box title "CODLogout".

```

-----

Una vez concluido el proceso de envío de la documentación firmada anterior, es posible conocer el "estado" y realizar un "seguimiento" del mismo de manera individual usando el "código" asignado a cada certificado de origen enviado con el webservice "**CODVerifica**".

Es posible también hacer un seguimiento de conjunto con los siguientes webservices:

**"CODTablaEnviados"**

Certificados de Origen aceptados en el sistema y en proceso de Aprobación o Rechazo por parte de un Funcionario de la Entidad. Siempre devuelve los Certificados de Origen que estén en este estado en los últimos 15 días (normalmente durante el día o al día hábil siguiente del envío un funcionario termina aprobando o rechazando, por lo cual es normal y frecuente que esta tabla contenga pocos datos del último día solamente, o venga directamente vacía)

**"CODTablaAprobados"**

Certificados de Origen Aprobados, pero NO Emitidos. Están en proceso de firma y emisión por parte del funcionario y la entidad. Devuelve los certificados en este estado para los últimos 31 días (normalmente no hay certificados de origen en este estado por más de 2 o 3 días)

**"CODTablaRechazados"**

Certificados de Origen Rechazados o NO Aprobados. Normalmente por problemas formales detectados por un funcionario habilitado de la entidad. Devuelve los certificados en este estado para los últimos 15 días.

**"CODTablaEmitidos"**

Certificados de Origen Emitidos y Vigentes. Pueden ser digitales (COD) o bien en Papel. Devuelve los certificados en este estado para los últimos 2 años.

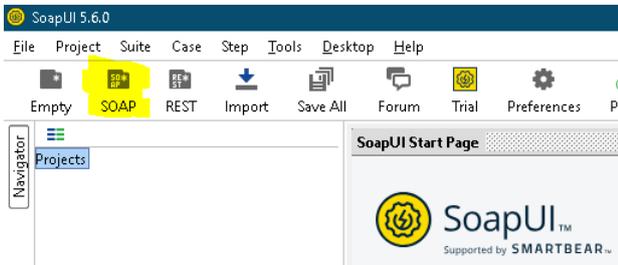
**"CODTablaAnulados"**

Certificados de Origen Emitidos y luego ANULADOS (o Anulados antes de la emisión formal). Pueden ser digitales (COD) o bien en Papel. Devuelve los certificados en este estado para los últimos 2 años.

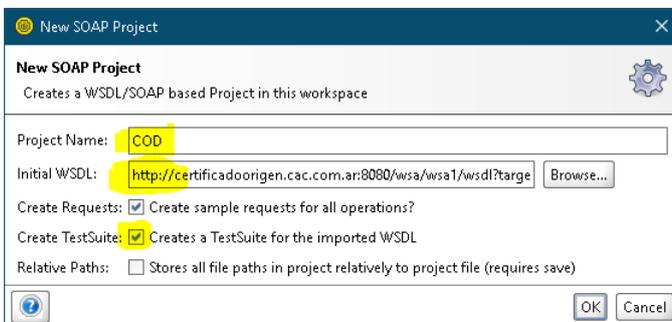
# ANEXO-1

## Ejemplo de uso de SOAP-UI

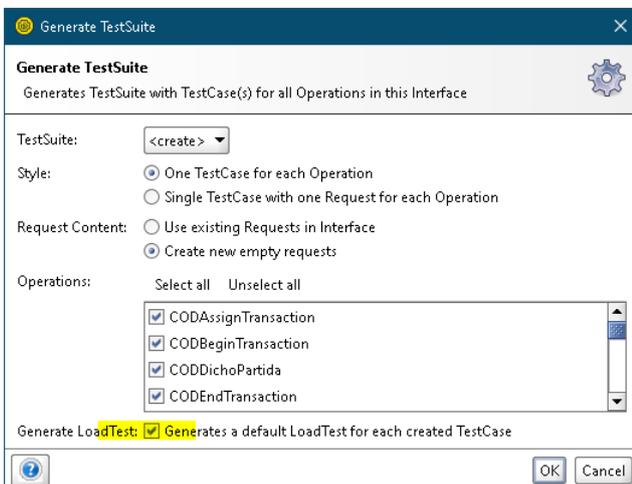
Creamos un nuevo proyecto SOAP



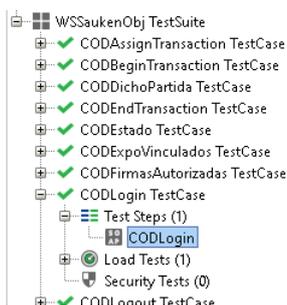
Asignamos WSDL <http://certificadoorigen.cac.com.ar:8080/wsa/wsa1/wsdl?targetURI=urn:localhost-org>



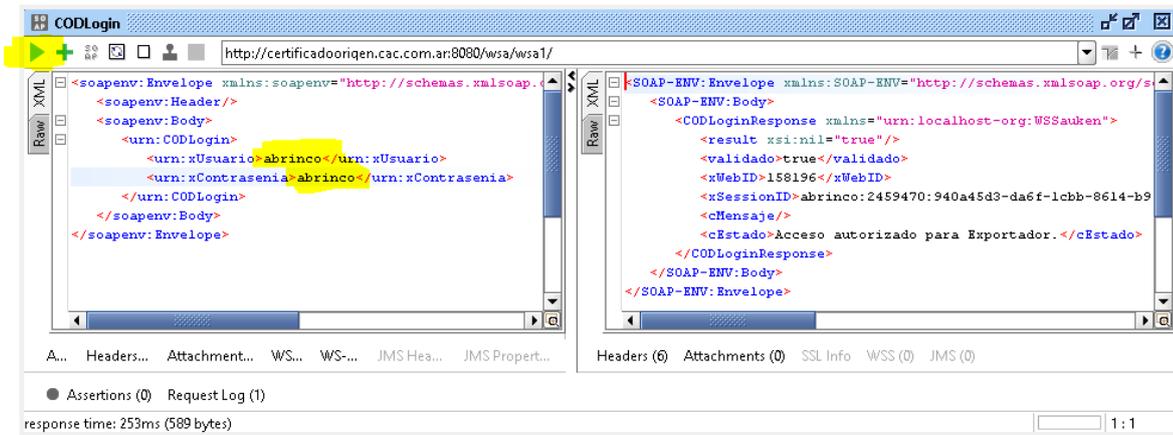
Generamos Test Suite y asignamos un nombre al proyecto



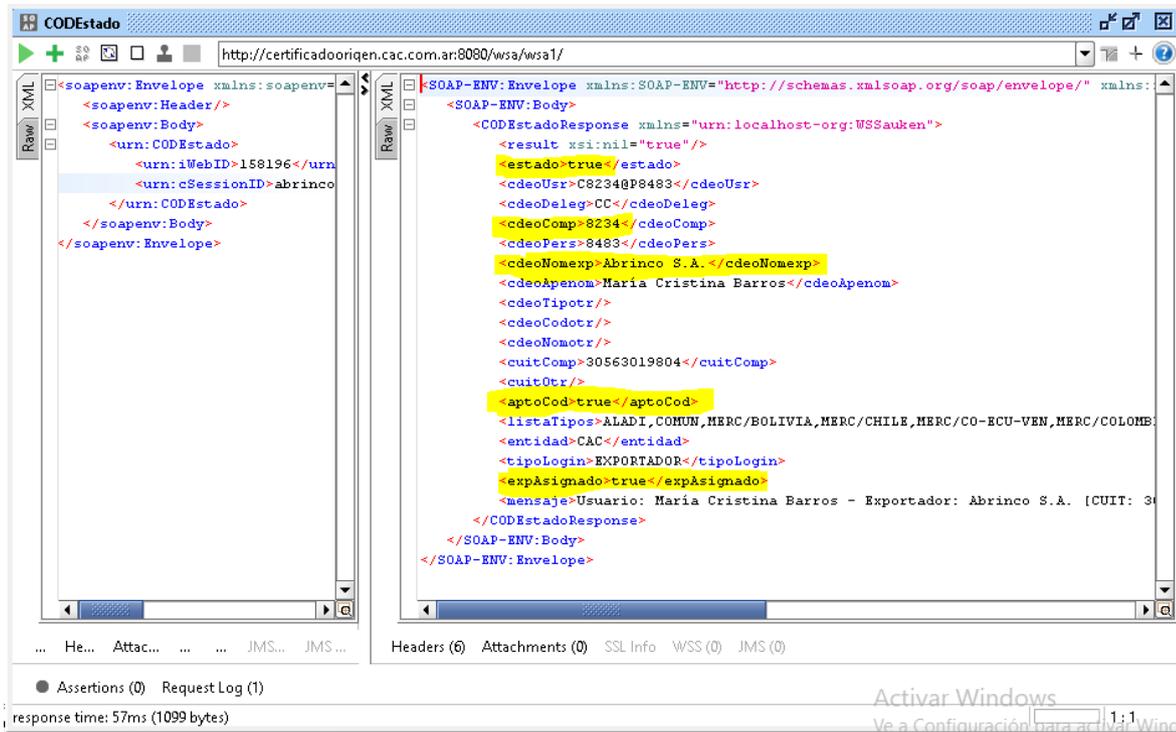
Probamos conectarnos con CODLogin



Cargar datos en XML de entrada y luego ejecutar. Observar el XML de salida en la ventana vecina.



Ejemplo: Ver CODEstado luego de CODLogin. Para realizar esto hay que tomar del XML de salida el WebID y el SessionID y copiarlos en los parámetros de entrada de CODEstado



## ANEXO-2

### Listado de Web Services SOAP

**/\*--- VISACIONES: CERTIFICADOS DE ORIGEN: CONTROL DE ACCESO Y SELECCIONES PARA EMPEZAR A OPERAR ---\*/**

CODLogin  
CODLogout  
CODSeleccionaExpo  
CODExpoVinculados  
CODFirmasAutorizadas

**/\*--- VISACIONES: CERTIFICADOS DE ORIGEN: TABLAS Y PROCESOS QUE DEVUELVEN PARÁMETROS ---\*/**

CODTablaPaíses  
CODTablaProvincias  
CODTablaUniMedida  
CODTablaTransportes  
CODTablaAcuerdos  
CODTablaNormasOrigen  
CODTablaNLCM  
CODTablaNALADI  
CODTablaNALADISA  
CODDichoPartida

**/\*--- VISACIONES: CERTIFICADOS DE ORIGEN: TRANSACCIONES COMPLEJAS ---\*/**

CODBeginTransaction  
CODEndTransaction

**/\*--- VISACIONES: CERTIFICADOS DE ORIGEN: ESTADOS ---\*/**

CODEstado  
CODValida  
CODVerifica

**/\*--- VISACIONES: CERTIFICADOS DE ORIGEN: DOCUMENTACIÓN FIRMADA ---\*/**

CODEXPUploadXML  
CODEXPUploadFactura  
CODEXPUploadDDJJ  
CODTablaCarga  
CODDatoCarga

**/\*--- VISACIONES: CERTIFICADOS DE ORIGEN: LISTAS DE CERTIFICADOS DE ORIGEN EN DISTINTOS PROCESOS/ESTADOS ---\*/**

CODTablaEnviados  
CODTablaAprobados  
CODTablaRechazados  
CODTablaEmitidos  
CODTablaAnulados

**/\*--- VISACIONES: CERTIFICADOS DE ORIGEN: DESCARGA DE CERTIFICADOS DE ORIGEN APROBADOS Y/O EMITIDOS ---\*/**

CODDownloadXMLEmitido  
CODDownloadPDFImpresion

**/\*--- VISACIONES: CERTIFICADOS DE ORIGEN: CONVERSIÓN DE ARCHIVOS Y PREPARACIÓN DE DATOS (no usados por el momento) ---\*/**

CODUploadCSV  
CODDownloadXMLExportador  
CODUploadFactura  
CODUploadDDJJ  
CODUploadXML  
CODDownloadPDFPre