Manual para el desarrollador

Autenticación de Servicios Web con Clave Ciudad

V1.6

Tabla de contenidos

Revisión Histórica

Versión	Fecha	Edición	Descripción
1.0	19/05/2017		Creación del documento
1.1	16/08/2017		Modificación del documento
1.2	31/08/2017		Modificación del documento
1.3	07/09/2017		Modificación del documento
1.4	04/10/2017		Modificación del documento
1.5	20/10/2017		Modificación del documento
1.6	06/11/2017		Modificación del documento

Introducción

Breve descripción – Objetivos

En este documento se definen los aspectos más importantes del proceso de utilización de servicios web en Clave Ciudad.

Conceptos Básicos:

Servicio:

Es el servicio al cual se intenta dar un acceso autenticado, este puede estar desarrollado en cualquier tecnología, por ejemplo RES o SOAP, el único requisito que posee es implementar los mecanismos de validación de la autenticación que se detallan en este documento, por lo cual deberá tener un parámetro extra pare este fin.

Clientes:

Los clientes son aplicaciones de software desarrolladas por terceros que utilizan los servicios.

Usuario:

El usuario o representado es la persona física o jurídica que quiere hacer uso del servicio a través del Cliente y que obligatoriamente posee Clave Ciudad. Cuando el usuario es una persona jurídica obligatoriamente operará a través de un representante, esto es opcional en el caso que usuario sea una persona física.

Representante:

Es la persona física responsable de crear los **Alias** de sus representados.

Alias:

Es una función que está incluida dentro de la aplicación Administración de Certificados que permite efectuar la administración de los accesos a los distintos Clientes. Con el Alias se relacionan los servicios generando de esta forma la autorización de acceso. Es recomendable generar un Alias por Cliente para que cada uno tenga acceso sólo a los servicios asignados.

Por ejemplo, el representante tiene un sistema de facturación que accede a servicios del e-Arciba y además un sistema de gestión interna que utiliza los servicios de “Trámites a Distancias (TAD)”. Para evitar que los sistemas pudieran tener un acceso indebido, el representante deberá tener dos Alias y darle acceso a cada uno.

Administrador de Certificados:

Es designado por el Representante en el momento que define el Alias. Esta asignación le permitirá agregar un certificado con el que accederá al servicio correspondiente al Alias. Esta también puede ser función del Representante

Token:

Parámetro que contiene toda la información sobre la autenticación, información sobre el usuario y el representante, tiempo de expiración, etc.

Sign:

Parámetro que sirve para validar que la información contenida dentro del token fue generada por la AGIP y permite verificar que no fue adulterado su contenido por un tercero

Certificado Digital:

Para poder acceder a un servicio, el cliente debe presentar un certificado de seguridad que fue previamente generado en el sitio de la AGIP y determina tanto al usuario como al representante. Los certificados deben estar asociados a un alias y tienen un tiempo de vigencia provisto por la AGIP.

LoginWS:

Es un servicio web SOAP publicado por la AGIP cuyo objetivo es que el cliente realice el pedido de autenticación. En el llamado se envía el certificado digital y el nombre del servicio al que se desea acceder.

En caso de error en la autenticación, se devolverá un mensaje detallado con un código de error. Si la autenticación fue correcta se devuelve un token y sign con toda la información que necesita el cliente. El token tiene una validez de 12 hs., por lo que el cliente podrá guardarlo y reutilizarlo durante ese tiempo todas las veces que lo necesite.

Secuencia de autenticación y ejecución (Cliente):

El cliente llama al LoginWS y obtiene el token y sign que le servirá por las próximas 12 hs.

El cliente llama al servicio enviándole la información propia del mismo más el token y sign provistos por LoginWs

Vencidas las 12 hs., el cliente vuelve a llamar a LoginWs para poder continuar con las próximas ejecuciones del servicio.

Descripción General del Servicio

El WS de Autenticación de Servicios Web con CC (LoginWS) es un servicio B2B ("Business to Business") que permite que los computadores pertenecientes a la AGIP y Entes Externos a la AGIP intercambien información en forma directa sin intervención de operadores. En dicha tarea intervienen los siguientes componentes:

- Un cliente de WS desarrollado por un EE siguiendo las especificaciones de este documento.
- El WS publicado por la AGIP que implementa la autenticación de los computadores del EE (CEE) mediante certificados digitales X.509 y la autorización del mismo como consumidor de un determinado WebService de Negocio (WSN).

Al usar especificaciones y protocolos estándares (PKI, XML, CMS, WSDL y SOAP) el cliente puede ser desarrollado con cualquier lenguaje de programación moderno.

Para que un Ente Externo a la AGIP (EE) esté autorizado a usar un WSN de AGIP, deberá realizar un trámite administrativo previo, cuya descripción esta fuera del alcance de este documento. Una vez finalizado exitosamente dicho trámite, el que incluye el alta de los CEE, el EE quedará registrado en el servicio de autorización de AGIP como entidad autorizada para usar el WSN.

Para que un CEE pueda utilizar efectivamente un WSN, deberá solicitar un "Ticket de Acceso" (TA) por medio del WS de Autenticación de Servicios Web CC (LoginWS). Dicho requerimiento se realiza mediante el envió de un "Ticket de Requerimiento de Acceso" (TRA) del CEE al LoginWS, mediante mensajería SOAP.

El LoginWS realiza la verificación del TRA y si el requerimiento es correcto, devuelve un mensaje que contiene el TA que habilita al CEE a utilizar el WSN solicitado. El TA deberá ser utilizado por el CEE para acceder al WSN.

En términos generales, el presente documento detalla las operaciones a realizar para:

- Generar un "Ticket de Requerimiento de Acceso" (TRA)
- Invocar el "Web Service de Autenticación y Autorización" (LoginWS)
- Interpretar el mensaje de respuesta del LoginWS y obtener el "Ticket de Acceso" (TA)

Referencias

Para mejor entendimiento de la presente especificación, se recomienda estar familiarizado con los siguientes estándares:

- PKI, http://www.pki.org
- XML, http://www.w3.org/TR/XML/
- SOAP, http://www.w3.org/TR/soap/
- WSDL, http://www.w3.org/TR/wsdl/
- CMS, http://www.ietf.org/rfc/rfc3852.txt
- NTP, http://www.ntp.org

Invocación del LoginWs

Sincronización de Clocks:

La fecha y hora del computador deberá estar sincronizada a través del protocolo NTP.

WSDL del LoginWs:

A continuación, se expone el WSDL perteneciente al LoginWS. El mismo estará disponible en una URL de la AGIP.
Homologación: https://hml.agip.gob.ar/claveciudad/webservice/LoginWS?wsdl
Producción: https://lb.agip.gob.ar/claveciudad/webservice/LoginWS?wsdl

Flujo Principal:

A continuación se describen los pasos que se deberán seguir para solicitar un TA al LoginWS. Cada uno de los puntos es explicado detalladamente en los apartados siguientes.

1. Generar el mensaje del TRA (LoginTicketRequest.xml)
2. Generar un CMS que contenga el TRA, su firma electrónica y el certificado X.509 (LoginTicketRequest.xml.cms)
3. Codificar en Base64 el CMS (LoginTicketRequest.xml.cms.bse64)
4. Invocar LoginWS con el CMS y recibir LoginTicketResponse.xml
5. Extraer y validar la información de autorización (TA).

Generación del documento del TRA (LoginTicketRequest.xml):

El primer paso para solicitar un TA es preparar el documento del TRA (denominado LoginTicketRequest.xml). Se puede utilizar una estructura XML ya definida que puede ser obtenida de un archivo externo o declarada como constante en el propio código. El esquema (schema, XSD) que describe dicho XML es el siguiente:

```
<?xml version="1.0" encoding="UTF8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:element name="loginTicketRequest" type="loginTicketRequest" />
 <xsd:complexType name="loginTicketRequest">
 <xsd:sequence>
 <xsd:element name="header" type="headerType" minOccurs="1" maxOccurs="1"/>
 <xsd:element name="service" type="serviceType" minOccurs="1"maxOccurs="1"/>
 </xsd:sequence>
 <xsd:attribute name="version" type="xsd:decimal" use="optional"default="1.0" />
 </xsd:complexType>
 <xsd:complexType name="headerType">
 <xsd:sequence>
 <xsd:element name="source" type="xsd:string" minOccurs="0" maxOccurs="1" />
 <xsd:element name="destination" type="xsd:string" minOccurs="0" maxOccurs="1"/>
 <xsd:element name="uniqueId" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
 <xsd:element name="generationTime" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
<xsd:element name="expirationTime" type="xsd:dateTime" minOccurs="1" maxOccurs="1" />
</xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="serviceType">
<xsd:restriction base="xsd:string">
<xsd:pattern value="[a-z,A-Z][a-z,A-Z,\-,_,0-9]*"/>
```

```
<xsd:minLength value='3'/>
<xsd:maxLength value='32'/>
</xsd:restriction>
</xsd:simpleType>
</xsd:schema>
```

A continuación se detalla la descripción de los atributos. Los mismos deben respetar el formato definido en el XSD:

- ***source***: Campo opcional. Indica el DN del certificado que será utilizado por LoginWS para verificar la firma electrónica del TRA generado por el computador (CEE) que realiza el requerimiento. Si no se incluye, se utilizará el primer certificado de firma incluido en el CMS. Si se incluye, deberá corresponder a uno de los certificados de firma incluidos en el CMS.
- ***destination***: Campo opcional. Indica el DN del LoginWS, En caso de utilizarse, deberá ser "C=ar,O=GCBA,CN=AGIP,serialNumber=CUIT 34999032089" tanto para el ambiente de homologación como de producción.
- ***uniqueId***: Entero de 32 bits sin signo que junto con "***generationTime"*** identifica el requerimiento.
- ***generationTime***: Momento en que fue generado el requerimiento. La tolerancia de aceptación será de hasta 24 horas previas al requerimiento de acceso.
- ***expirationTime***: Momento en el que expira la solicitud. La tolerancia de aceptación será de hasta 12 horas posteriores al requerimiento de acceso.
- ***service***: Identificación del WSN para el cual se solicita el TA.

Consulte en la documentación el nombre del servicio correspondiente.

El siguiente es un ejemplo del documento LoginTicketRequest.xml generado por la empresa SA cuya CUIT es 30123456789 y el DN del CEE es cn=srv1,o=empresa s.a.,c=ar,serialNumber =CUIT 30123456789 20111111110 27111111110 solicitando acceso al WSN wsfe:

```
<?xml version="1.0" encoding="UTF8"?>
<loginTicketRequest version="1.0">
<header>
     <uniqueId>4325399</uniqueId>
     <generationTime>2017-11-02T10:00:00</generationTime>
     <expirationTime>2017-11-02T19:10:00</expirationTime>
</header>
<service>NOMBRE_SERVICIO</service>
</loginTicketRequest>
```

Generación del Ticket de Requerimiento de Acceso (TRA)

Se deberá generar un mensaje CMS del tipo "SignedData" que contenga el mensaje anteriormente generado (LoginTicketRequest.xml) y su firma electrónica utilizando SHA1+RSA. De esta forma, se obtiene el TRA (LoginTicketRequest.xml.cms).

Codificación en Base64 el TRA:

Para poder enviar el TRA al LoginWS, el mismo deberá ser codificado en Base64 (LoginTicketRequest.xml.cms.bse64)

Envió del TRA al LoginWS:

Se debe invocar al método getLoginTicketFromCMS del LoginWS.
El mismo recibe como parámetro una cadena correspondiente a la codificación en Base64 del TRA (LoginTicketRequest.xml.cms.base64) y devuelve una cadena denominada LoginTicketResponse.xml.
De esta última se deberá extraer el Ticket de Acceso (TA).
También está disponible para usuarios que tienen problemas generando el objeto que devuelve getLoginTicketFromCMS, el método getLoginTicketFromCMS_STR que devuelve un string con la misma información que el método del primer párrafo.

Mensajes de Error:

En caso de encontrarse algún error, el mensaje SOAP devolverá un "SoapFault" conteniendo el código y descripción del error producido. La descripción podrá contener adicionalmente detalles más específicos del error (ej: el XML expiró hace 10 minutos). La siguiente tabla lista los códigos de errores y su correspondiente descripción. En caso de que la AGIP considere necesario, nuevos códigos de errores y su descripción serán agregados.

Código	Descripción
50	No fue valido el CMS
51	No se pudo obtener la firma del CMS
52	No se pudo obtener el certificado del CMS
53	La firma del CMS no es válida.
54	El certificado no fue firmado por la AGIP.
55	El CMS no posee firma.
56	No se encontró la firma que se corresponde con el source indicado.
57	El CMS no posee certificado para la firma.

58	No se encontró el certificado que se corresponde con el source indicado.
59	Formato inválido del XML loginTokenRequest.
60	No se admite un GenerationTime futuro.
61	No se admite un GenerationTime mas antiguo de 24hs.
62	No se admite un ExpirationTime ya expirado.
63	No se admite un ExpirationTime de mas de 24hs.
64	Certificado no encontrado en AGIP
65	Serialnumber del certificado inválido.
66	Certificado distinto al firmado por AGIP
67	No se encontró el servicio o no se tiene acceso al mismo con el alias.
68	Servicio indicado no es un servicio web
69	Servicio deshabilitado
70	El contribuyente debe inscribirse en DFE
71	uniqueId duplicado.
72	Debe especificar un header.
73	Debe especificar un GenerationTime.
74	Debe especificar un ExpirationTime.
75	La firma de la AGIP no pudo ser verificada.
76	No pudo ser leído el CMS.
77	Acceso denegado debido al cambio de administrador
78	Certificado expirado
79	Alias desactivado
11000	Error interno del sistema

Extracción y validación del TA

LoginTicketResponse.xml es descripto en el siguiente esquema (schema, XSD):

```
<?xml version="1.0" encoding="UTF8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="loginTicketResponse" type="loginTicketResponse" />
<xsd:complexType name="loginTicketResponse">
<xsd:sequence>
<xsd:element name="header" type="headerType" minOccurs="1" maxOccurs="1" />
<xsd:element name="credentials" type="credentialsType" minOccurs="1" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="version" type="xsd:decimal" use="optional" default="1.0" />
</xsd:complexType>
```

```
<xsd:complexType name="headerType">
<xsd:sequence>
<xsd:element name="source" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="destination" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="uniqueId" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1" />
<xsd:element name="generationTime" type="xsd:dateTime" minOccurs="1" maxOccurs="1" />
<xsd:element name="expirationTime" type="xsd:dateTime" minOccurs="1" maxOccurs="1" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="credentialsType">
<xsd:sequence>
<xsd:element name="token" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="sign" type="xsd:string" minOccurs="1" maxOccurs="1" />
</xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

Los datos incluidos son los siguientes:

• ***source***: DN correspondiente al LoginWS que generó el documento (producción / homologación)

• ***destination***: DN correspondiente del CEE autenticado por el LoginWS

• ***uniqueId***: Entero de 32 bits sin signo que junto a "***generationTime"*** identifica al requerimiento.

• ***generationTime***: Momento en que fue generado el TA.

• ***expirationTime***: Momento en el que expira el TA.

• ***token*** y ***sign***: cadenas de caracteres que deben ser informadas al WSN (como variables TOKEN y SIGN). Las mismas componen el TA. El formato interno de estas cadenas puede diferir de un servicio a otro y su información contenida es interpretada por el WSN.

Se deberá verificar que el mensaje de respuesta, que incluye al TA, no se encuentre expirado mediante la variable "expirationTime" y su momento de generación sea válido mediante la variable "generationTime".

Un ejemplo de respuesta al requerimiento expuesto anteriormente es el siguiente:

```
<loginTicketResponse version="1.0">
<credentials>
<sign>EG6okj3D/su9L3sn5H7gi+TcasGczZNM6c11+iTSHse3ilMRaLfIUriISwyRQKqBeJmXjUD53xxwmQ1BVTtND3BzZ
yl9o=</sign>
<token>PD94bWwgdmVyZm9ubz0iIiBlbWFpbD0iIiB0aXBvRG9jdW1lbnRvPSIwMDM5OTkiIGVsZWdpZG89InRydWUiLz48
L3JlcHJlc2VudGFkb3M+PC9kYXRvcz4=</token>
</credentials>
<header>
<destination>C=AR,O=organismo,CN=nombre,serialNumber=CUIT 30334445556 23216668889
20998887775</destination>
<expirationTime>2017-11-03T22:46:57.071-03:00</expirationTime>
<generationTime>2017-11-03T10:46:57.071-03:00</generationTime>
<source>C=ar,O=GCBA,CN=AGIP,serialNumber=CUIT 34999032089</source>
<uniqueId>3095</uniqueId>
</header>
</loginTicketResponse>
```

Notar que el tiempo de vida del TA presente en este ejemplo es de 12 horas. El CEE podría utilizar este TA sin necesidad de solicitar otro, indistintamente de la cantidad de veces que consuma el servicio al cual solicito acceso. Para el acceso a un WSN para el cual un CEE posea un TA valido, se recomienda utilizar dicho TA y no solicitar uno nuevo.

Requerimientos de los certificados pertenecientes a los COE:

La autenticación de los CEE, se realizara mediante certificados "X.509v3". Los mismos deberán cumplir con los siguientes requerimientos:

1. Ser emitido por una autoridad certificante reconocida por AGIP.
2. El DN deberá enmarcarse dentro de la RFC 2253 (http://www.ietf.org/rfc/rfc2253.txt)
3. El contenido del DN se debe cumplir los siguientes requisitos establecidos por la "Oficina Nacional de Tecnologías de Información" (ONTI), disponible en: http://www.sgp.gov.ar/contenidos/onti/productos/docs/infraestructura/Anexo_III_Perfil_Minimo_de_Certificados_y_CRLs_v1.pdf
4. Los campos obligatorios son los siguientes:
   - Campo 'commonName': DEBE corresponder al nombre del servicio o aplicación (ej. Sistema de Consulta) o al nombre de la unidad operativa responsable del servicio (ej. Gerencia de Compras).
   - Campo 'serialNumber' (OID 2.5.4.5: Nro de serie): DEBE contener el número de identificación de la Persona Jurídica Pública o Privada, expresado como texto y respetando el siguiente formato y codificación: "CUIT numero_de_cuit1 numero_de_cuit2 numero_de_cuit3"
   (el primer CUIT corresponde al representado, el segundo al representante y el tercero al administrado)

   - Campo 'organizationName': DEBE coincidir con el nombre de la Persona Jurídica Pública o Privada.
   - Campo 'countryName': DEBE representar el país en el cual está constituida la Persona Jurídica, codificado según el estándar [ISO3166].

Secuencia completa de comandos con openssl para generar el token/sign y testeo mediante SoapUI:

1. Genera "csr" con la clave privada

openssl req -new -key privada.key -subj "/C=AR/O=organismo/CN=nombre/serialNumber=CUIT 30121231231 20112223331 23998887776" -out archivo.csr

2. Obtener certificado

Para el ambiente de producción, entrar mediante Clave Ciudad al aplicativo de "Webservices" para la "Administración de Certificados", seguir los pasos de la ayuda para descargar el certificado.

3. Pasar a formato "pem"

openssl x509 -inform der -in certificado.crt -out certificado.pem

4. Genera "CMS", este comando ya lo deja en base64

openssl smime -sign -signer certificado.pem -inkey privada.key -out archivo.cms -in LoginTicketRequest.xml -outform PEM -nodetach

```
EJ: LoginTicketRequest.xml :
<?xml version="1.0" encoding="UTF8"?>
<loginTicketRequest version="1.0">
<header>
    <uniqueId>43259</uniqueId>
    <generationTime>2017-11-02T10:00:00</generationTime>
    <expirationTime>2017-11-02T19:10:00</expirationTime>
</header>
<service>NOMBRE_SERVICIO</service> <<<<==== REEMPLAZAR POR EL NOMBRE CORRECTO QUE UD REQUIERA!!!!!!
</loginTicketRequest>
```

5. Test con SoapUI, ENVIO

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soap="http://soap.controller.cc.agip.gov.ar/">

   <soapenv:Header/>
   <soapenv:Body>
      <soap:getLoginTicketFromCMS>
      <CMS> **** COLOCAR AQUI EL RESULTADO DEL PASO ANTERIOR****</CMS>
         </soap:getLoginTicketFromCMS>
   </soapenv:Body>
</soapenv:Envelope>
```

SoapUI, DEVUELTO

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:getLoginTicketFromCMSResponse xmlns:ns2="http://soap.controller.cc.agip.gov.ar/">
         <loginTicketResponse version="1.0">
            <credentials>

<sign>EG6okj3D/su9LBUMvpw/N8eGgqaxoZ+e3ilMRaLfIUriISwyRQKqBeJmXjUD53xxwmQ1BVTtND3BzZyl9o=</sign>

<token>PD94bWwgdmVyc2lviIiBlbWFpbD0iIiB0aXBvRG9jdW1lbnRvPSIwMDM5OTkiIGVsZWdpZG89InRydWUiLz48L3Jlc
HJlc2VudGFkb3M+PC9kYXRvcz4=</token>
            </credentials>
            <header>
               <destination>C=AR,O=ORGA,CN=NOMBRE,SERIALNUMBER=CUIT 30123456789 23123456789
20123456789</destination>
               <expirationTime>2017-11-03T22:46:57.071-03:00</expirationTime>
               <generationTime>2017-11-03T10:46:57.071-03:00</generationTime>
               <source>C=ar,O=GCBA,CN=AGIP,serialNumber=CUIT 34999032089</source>
               <uniqueId>3095</uniqueId>
            </header>
         </loginTicketResponse>
      </ns2:getLoginTicketFromCMSResponse>
   </S:Body>
</S:Envelope>
```