Plataforma de Interoperabilidad

Tutorial .NET

Control de Cambios

Fecha	Versión	Descripción	Autor	Aprobado Por
2015	1.0	Versión inicial		

Nombre actual del archivo: AGESIC-Plataforma-Tutorial-dotNET-v01-00.odt

Plataforma de Interoperabilidad

Tutorial: Consumir un servicio sincrónico de la PGE

Objetivo

El objetivo de este tutorial es proveer una guía paso a paso para el desarrollo de un cliente *desktop* de la Plataforma de Gobierno Electrónico (PGE) sobre la plataforma .NET.

Prerrequisitos

Se asume que el usuario conoce las especificaciones WS-Security, WS-Trust, SAML 1.1. Además, se asume que el usuario está familiarizado con el uso de certificados, aplicaciones .NET y Web Services.

Se debe haber completado anteriormente el tutorial: Certificados_Microsoft.

Requerimientos del software

La tabla 1 presenta las herramientas y productos de *software* requeridos para desarrollar y ejecutar la Aplicación Cliente.

Producto	Versión
.NET Framework	3.5
Visual Studio Express	2008

Tabla 1 – Requerimientos de Software

Descripción de un escenario

La figura 1 presenta el escenario de ejemplo que se utiliza en este tutorial, en el cual intervienen dos organismos: el Banco de Previsión Social (BPS) que será el Organismo Cliente (quien consume el servicio) y AGESIC que será el Organismo Proveedor.

AGESIC provee el servicio "**Timestamp**" el cual tiene una unica operaciòn "**GetTimestamp**"**.** Cuando se registró el servicio en la PGE, se creó un Servicio Proxy para que las Aplicaciones Cliente accedan al servicio a través de él (los clientes se comunican con el proxy y éste transfiere la invocación al servicio final; luego toma la respuesta de este último y la reenvía al cliente). Además, mediante la configuración de políticas de control de acceso, el AGESIC autorizó a los usuarios con rol "gerencia de proyectos" de la sección "agesic" (ou=gerencia de proyectos,o=agesic) a consumir el método "GetTimestamp" [1].

[1] Los roles autorizados a invocar una determinada operación de un servicio web son acordados entre el proveedor del servicio y AGESIC. Los clientes que deseen invocar cada operación deberán solicitar esta información a AGESIC.

Por otro lado, en el BPS hay una Aplicación Cliente que está siendo utilizada por el usuario Pruebas que tiene el rol mencionado. La aplicación necesita acceder al servicio de AGESIC para lo cual, utilizando las credenciales del usuario Pruebas y a través de una Aplicación Emisora de Tokens interna al BPS, obtiene un *token* de seguridad SAML firmado por el BPS (pasos 1.a y 1.b).

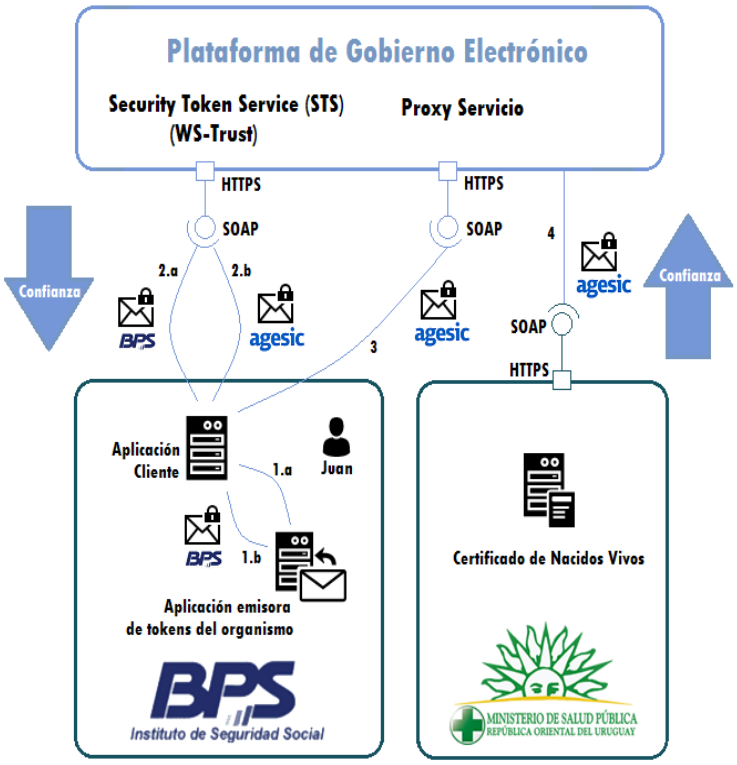


Figura *1*: Escenario de uso

Luego con el *token* recibido obtiene del STS de la PGE, utilizando el estándar WS-Trust, otro *token* de seguridad firmado por la plataforma (pasos 2.a y 2.b). Para emitir este *token* la PGE verifica la firma digital del *token* enviado por la aplicación y la existencia del rol “ou=gerencia de proyectos,o=agesic”.

Por último, la Aplicación Cliente invoca al Servicio del MSP a través del Servicio Proxy de la PGE (los clientes nunca acceden al servicio final directamente, siempre lo hacen a través del proxy creado en la PGE; existe un proxy específico para cada servicio disponible a través de la PGE). En la invocación se incluye el *token* firmado por la PGE y se especifican el servicio y el método a invocar.

Consumo de un servicio

En esta sección se describe paso a paso la implementación de una Aplicación Cliente .NET de escritorio. El servicio a consumir se llama Timestamp, y devuelve el fecha/hora de la PDI.

La implementación del escenario comprende las siguientes etapas:

- Crear proyecto .NET consola y agregar librerías de apoyo
- Crear una referencia al servicio
- Configurar WS-Addressing
- Configurar la conexión SSL
- Configurar comunicación con el STS de la PGE
- Invocación del Servicio

En las siguientes sub-secciones se describen en detalle cada una de ellas.

Crear proyecto .NET Consola y agregar librerías de apoyo

1. Seleccionar *File* → *New* → *Project* → *Visual C#* → *Console Application* y crear un proyecto con nombre ClienteTutorial y nombre de la solución *Tutorial* como se muestra en la figura 1.

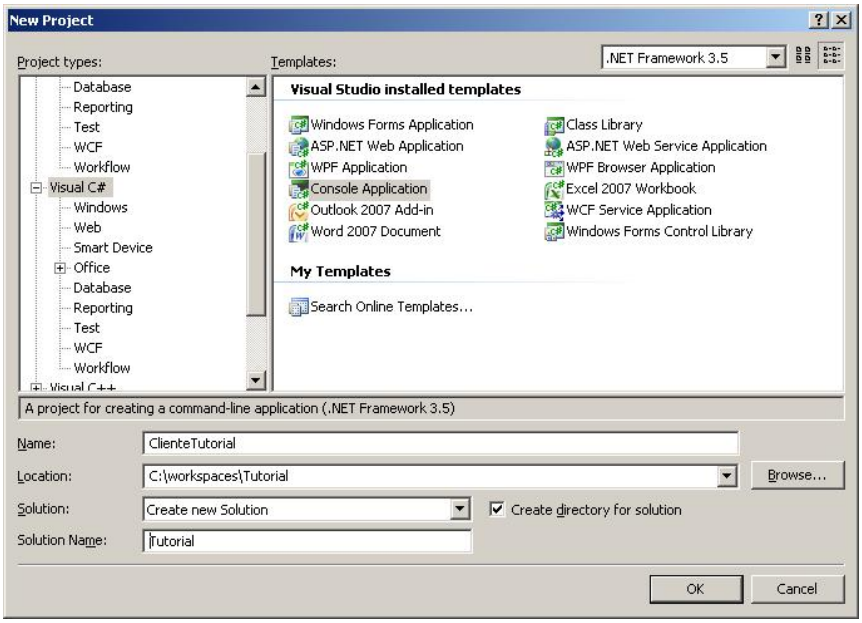


Figura 2: Creación de un proyecto .NET

2. Hacer clic derecho en la solución → *Add* → *Existing Project...* como se muestra en la figura 3.

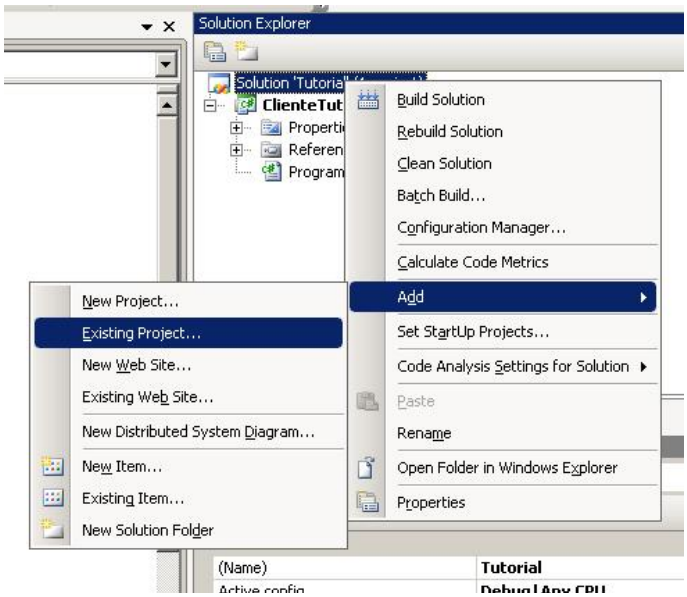


Figura 3: Agregar un proyecto existente

3. Seleccionar la ubicación del proyecto PGE extraído de la carpeta materiales bajada del FTP. El resultado esperado debe ser similar al de la figura 4.

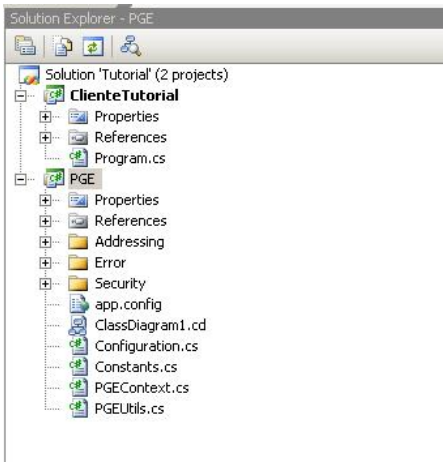


Figura 4: Proyecto agregado

4. Dentro del proyecto ***ClienteTutorial***, hacer clic derecho en *References* → *Add Reference...*, seleccionar la solapa *projects* y luego el proyecto PGE, como se muestra en la figura 5.

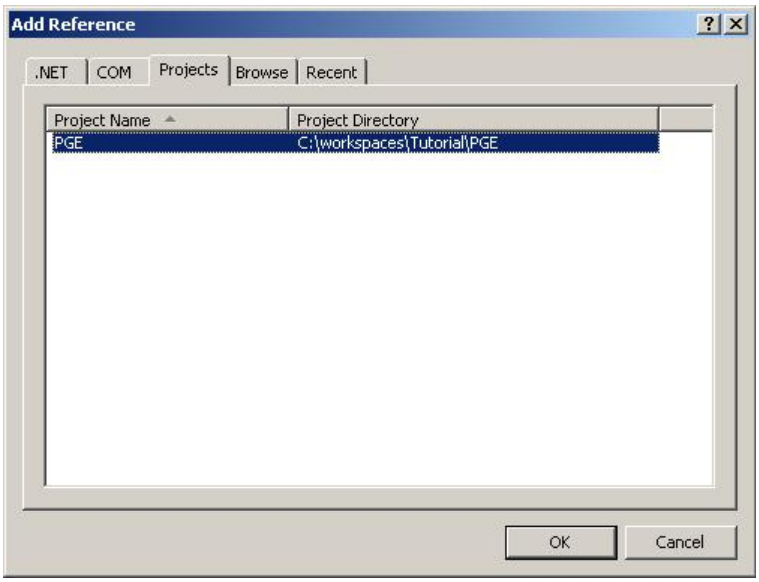


Figura 5: Agregar como referencia el proyecto PGE

Crear Referencia al Servicio

1. Hacer clic derecho en el proyecto ***ClienteTutorial*** y seleccionar *Add Service Reference...*

2. Especificar la dirección del WSDL del servicio como se muestra en la figura 6 y presionar el botón *Go*. El WSDL a seleccionar es TimestampServicewsdl.*wsdl*. (Carpeta WSDL)

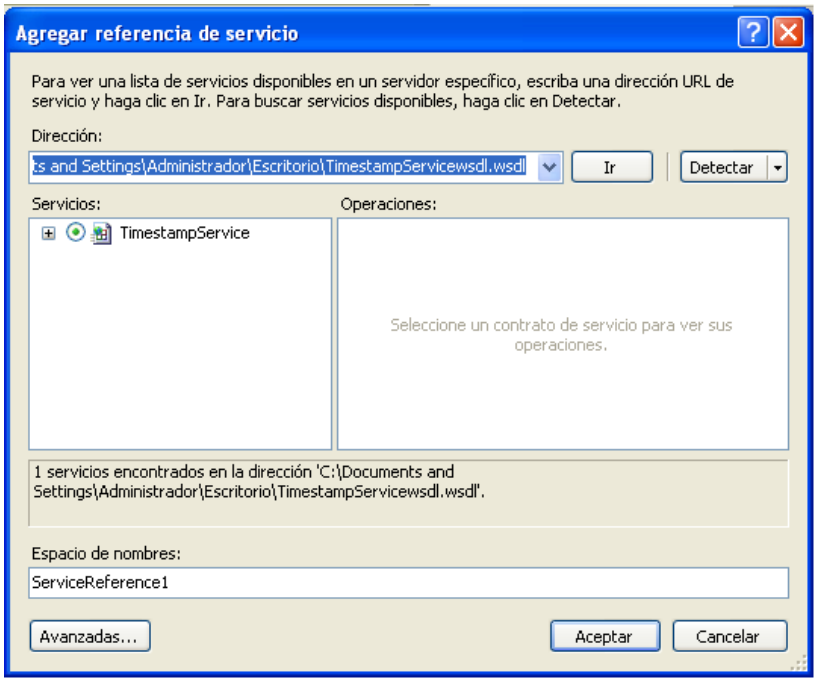


Figura 6: Agregar referencia al Web Service

3. Modificar el campo **namespace** a Tutorial y presionar el botón OK. Este paso creará un archivo de configuración del Web Service llamado *app.config.*

Configurar WS-Addressing

1. Hacer clic derecho en el archivo *app.config* y seleccionar *Edit WCF Configuration* como se muestra en la figura 7.

Nota: Debido a un bug en Visual Studio 2008, antes es de activar la herramienta gráfica es ncesario utilizar la opción *Edit WCF Configuration*. Para activarla, es ir al menú superior → *Tools* → *WCF Service Configuration Tool.*

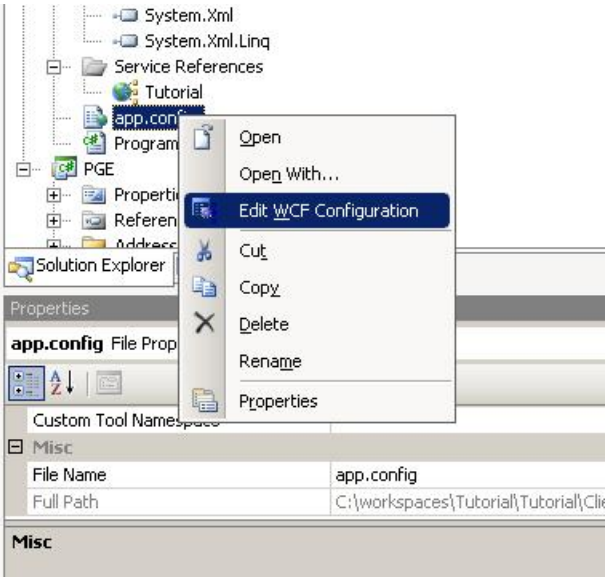


Figura 7: Editar el archivo de configuración del WS

2. Seleccionar *Advanced* → *Endpoint Behavior* → *New Endpoint Behaviour Configuration*

3. Nombrar al nuevo *behaviour* como *PGEBehaviour*

4. Presionar el botón *Add...*, luego client via como se muestra en la figura 8 y luego el botón *Add* nuevamente.

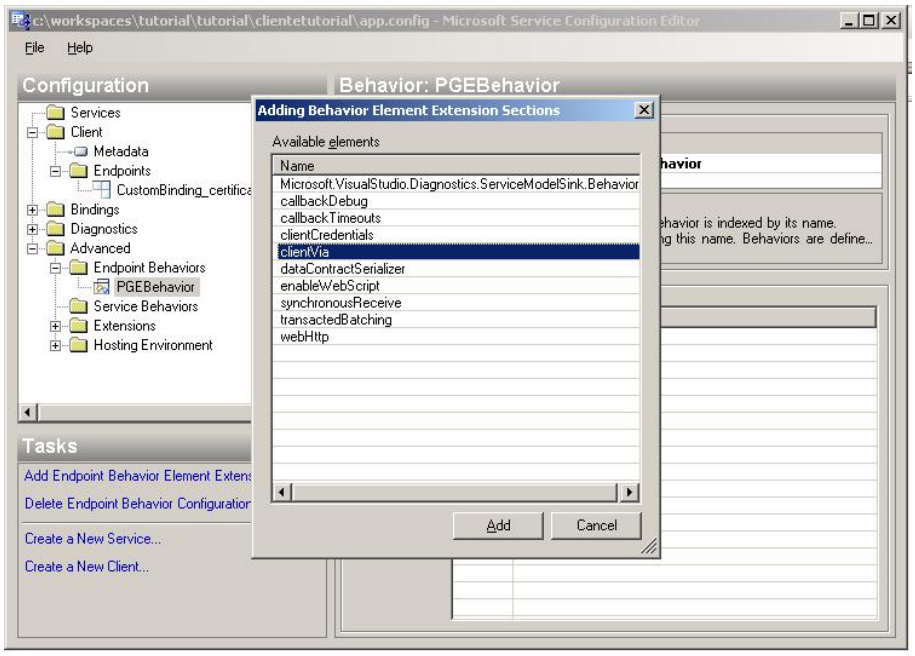


Figura 8: Definir el client via del endpoint

5. Definir *ViaUri* con el valor https://testservicios.pge.red.uy:6055/timestamp/TimestampService Este valor, representa la dirección física del servicio.

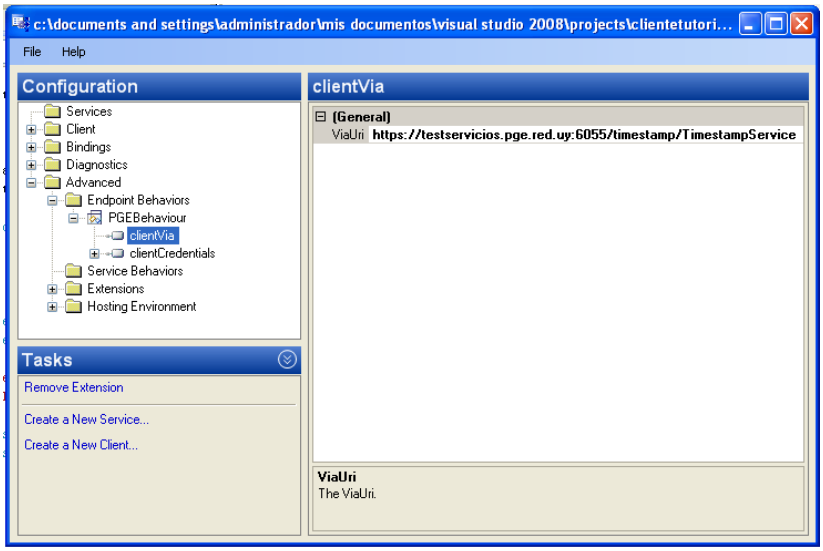
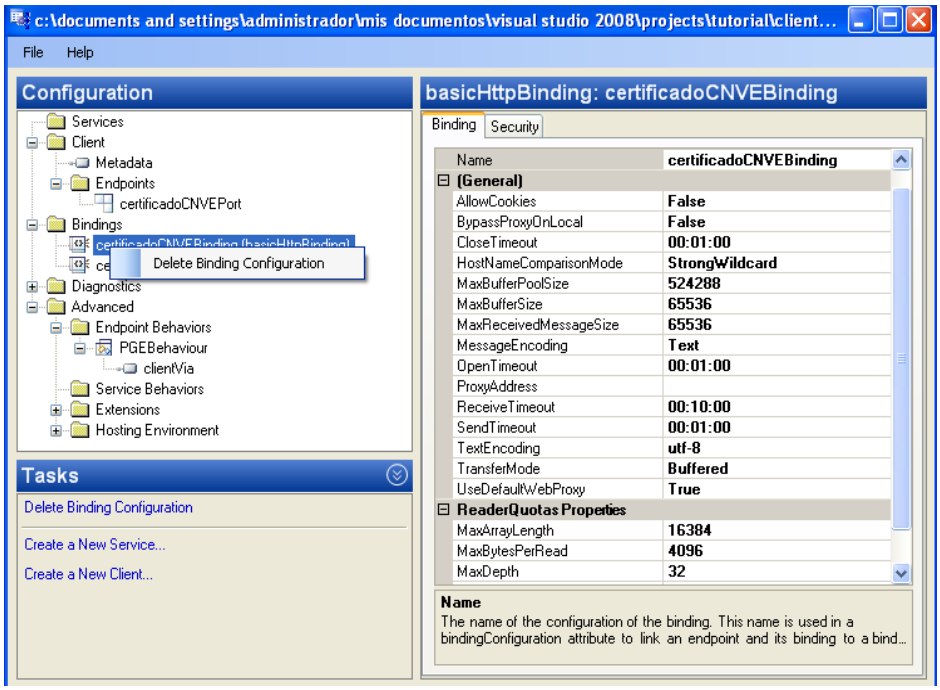


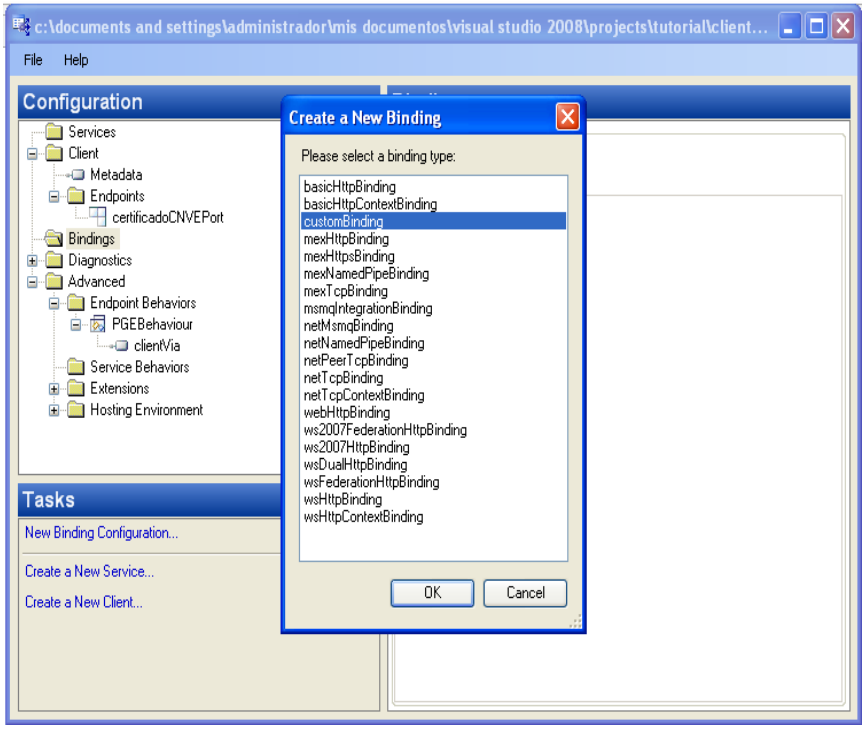
Figura 9: Especificar el valor ViaUri

Configurar la conexión SSL

1. En el menu de la izquierda, dentro de *Bindings*, eliminar cada uno de los bindings que se muestran:

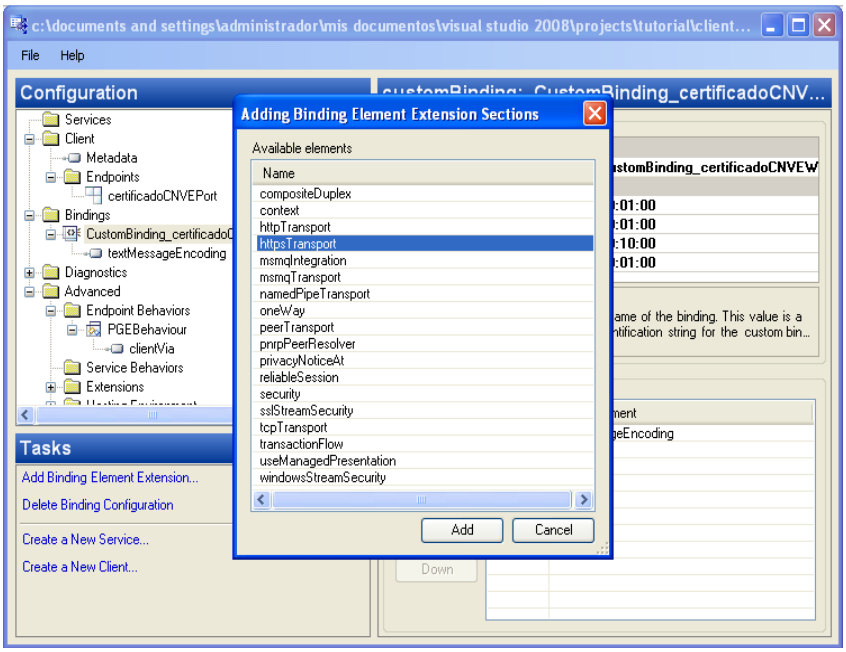


2. Hacer clic derecho en *Bindings* y seleccionar la opción “New Binding Configuration”. Seleccionar la opción “customBinding” y darle clic en OK.



3. Nombrar el nuevo *binding* de la siguiente manera: “*CustomBinding_TimestampWSDLPortType”*
4. Seleccionar abajo a la derecha el *Binding element “httpTransport”* y eliminarlo

5. Hacer clic en *"Add"* y seleccionar *"httpsTransport"*



6. Dentro de *httpsTransport*, seleccionar como true la opción *RequiredClientCertificate* como *true*, según se muestra en la figura 10.

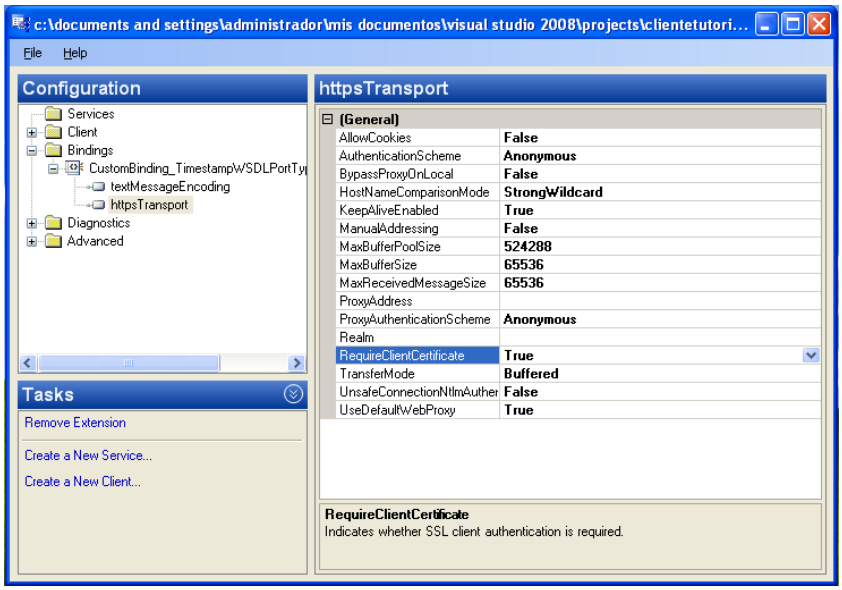


Figura 10: Configurar SSL para autenticación mutua

7. Hacer clic derecho nuevamente en *Bindings* → *CustomBinding_TimestampWSDLPortType* → *TextMessageEncoding* y verificar que el elemento *MessageVersion* tiene el valor *Soap11WSAddressing10.*

8. Seleccionar *Advanced → Endpoint Behaviours → PGEBehaviour*, luego clic derecho y seleccionar la opción *Add Endpoint Element Extension.*

9. Seleccionar la opción *clientCredentials*

10. Configurar el *clientCredentials → ClientCertificate* según muestra la figura 11.

    Aquí deberá poner el nombre del certificado propio que se le devolvió en el tutorial de certificados Windows, en lugar de "Agesic Testing".

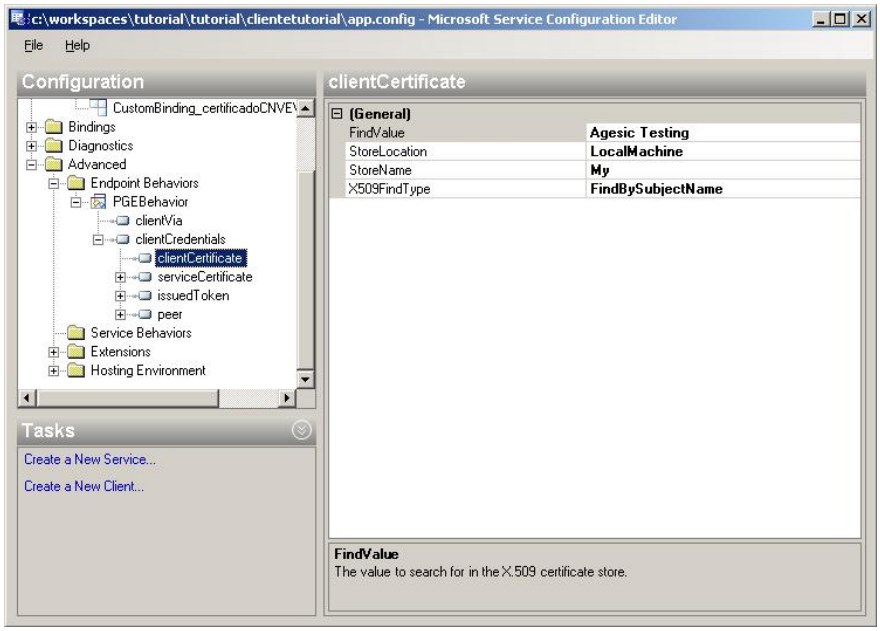


Figura 11: Configurar el certificado del cliente

11. Configurar el *clientCredentials → serviceCertificate → Default Certificate* según muestra la figura 12.

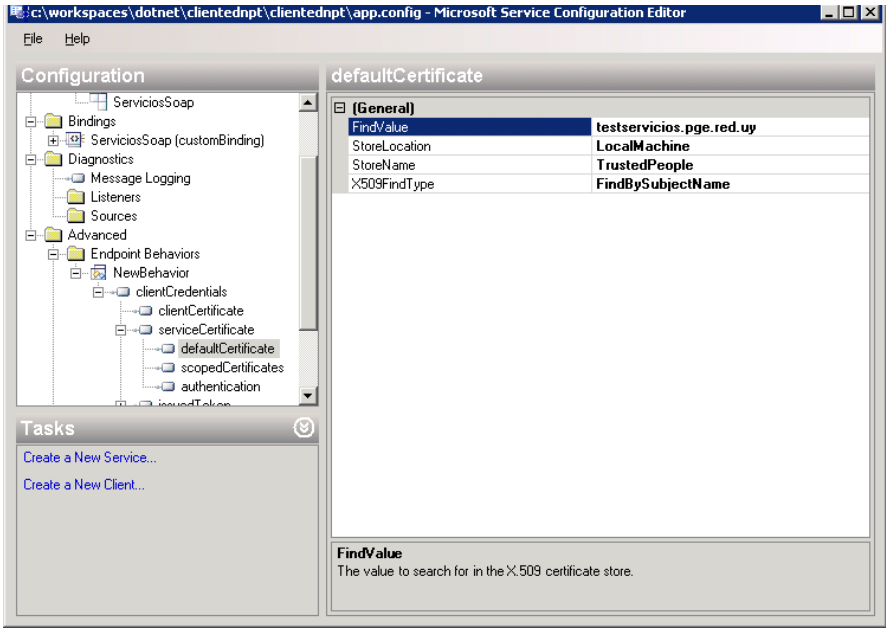


Figura 12: Configurar el certificado de la PGE

12. Configurar el *clientCredentials* → *serviceCertificate* → *authentication* según muestra la figura 13.

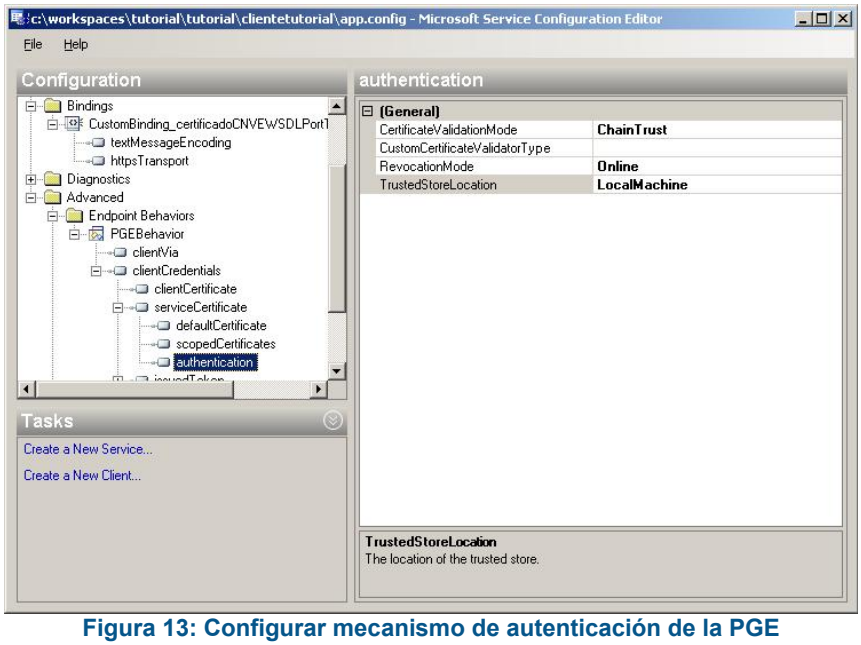
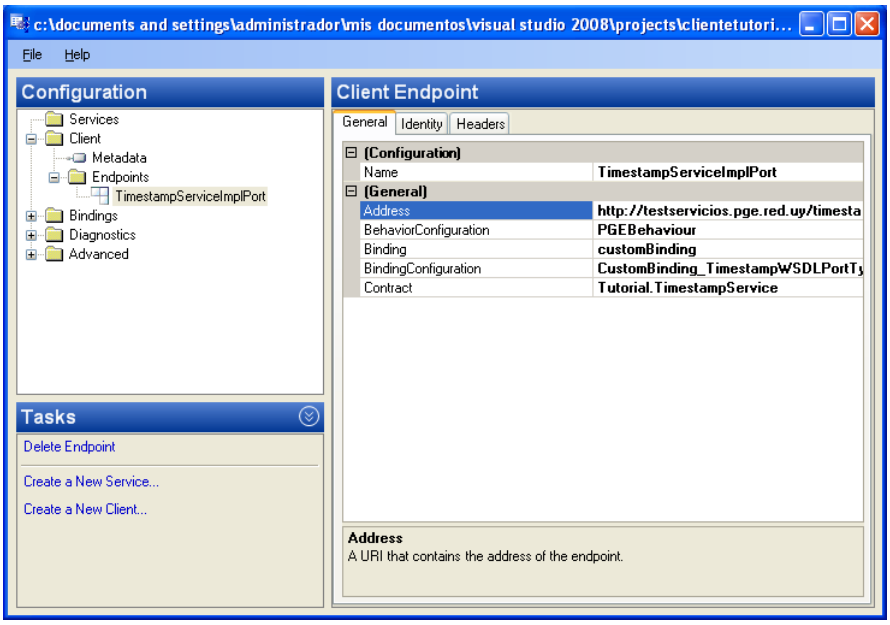


Figura 13: Configurar mecanismo de autenticación de la PGE

13. Asociar el PGE Behaviour creado al cliente. Para ello, seleccionar *Client* → *Endpoint* → *TimestampPort* → *BehaviourConfiguration* y seleccionar la opción *PGEBehaviour.* A su vez, en la opción *"Binding"* seleccionar *"customBinding"* y en *"Address" ingresar "http://testservicios.pge.red.uy/timestamp"*



14. Seleccionar *File* → *Save* y luego *File* → *Close*.

Configurar comunicación con el STS de la PGE

1. Abrir el archivo *app.config* y agregar el código en negrita de la figura 14 luego del tag *configuration* y antes del tag *system.serviceModel.*

```
<configuration>
  <configSections>
    <sectionGroup name="pgeConfigSectionGroup">
      <section name="PGEConfigSection" type="AGESIC.PGE.PGEConfigSection, Agesic.PGE,
Culture=neutral, Version=1.0.0.0"/>
    </sectionGroup>
  </configSections>
  <pgeConfigSectionGroup>
    <PGEConfigSection
    SAMLIssuer="AGESIC"
    STSUrl="http://testservicios.pge.red.uy:6001/TrustServer/SecurityTokenService"
    RSTCertificateDN="{{Colocar aquí el CN del certificado emitido, por ejemplo:
jperez.pge.red.uy}}"
    STSUrlSSL="https://testservicios.pge.red.uy:6051/TrustServer/SecurityTokenServiceProtected"
    SAMLMaxFrame="15"
    SAMLMinFrame="120"
    RSTCertificateStoreName="My"
    RSTCertificateStoreLocation="LocalMachine" />
  </pgeConfigSectionGroup>
  <system.serviceModel>
  ...
  </system.serviceModel>
</configuration>
```

Figura 14: Configurar opción del STS de la PGE

Con estos agregados, el archivo *app.config* se debería ver como se muestra a continuación:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
    <sectionGroup name="pgeConfigSectionGroup">
      <section name="PGEConfigSection" type="AGESIC.PGE.PGEConfigSection, Agesic.PGE,
&amp;#xD;&amp;#xA;Culture=neutral, Version=1.0.0.0"
                />
    </sectionGroup>
  </configSections>
  <pgeConfigSectionGroup>
    <PGEConfigSection
    SAMLIssuer="AGESIC"
    STSUrl="http://testservicios.pge.red.uy:6001/TrustServer/SecurityTokenService"
    RSTCertificateDN="{{Colocar aquí el CN del certificado emitido, por ejemplo: jperez.pge.red.uy}}"
    STSUrlSSL="https://testservicios.pge.red.uy:6051/TrustServer/SecurityTokenServiceProtected"
    SAMLMaxFrame="15"
    SAMLMinFrame="120"
    RSTCertificateStoreName="My"
    RSTCertificateStoreLocation="LocalMachine" />
  </pgeConfigSectionGroup>
    <system.serviceModel>
      <behaviors>
        <endpointBehaviors>
          <behavior name="PGEBehaviour">
            <clientVia viaUri="https://testservicios.pge.red.uy:6055/timestamp/TimestampService" />
            <clientCredentials>
              <clientCertificate findValue={{Colocar aquí el CN del certificado emitido, por ejemplo:
jperez.pge.red.uy}} storeLocation="LocalMachine"
                x509FindType="FindBySubjectName" />
              <serviceCertificate>
                <defaultCertificate findValue="testservicios.pge.red.uy" storeLocation="LocalMachine"
                  storeName="TrustedPeople" x509FindType="FindBySubjectName" />
                <authentication trustedStoreLocation="LocalMachine" />
              </serviceCertificate>
            </clientCredentials>
          </behavior>
        </endpointBehaviors>
      </behaviors>
      <bindings>
        <customBinding>
          <binding name="CustomBinding_TimestampWSDLPortType">
            <textMessageEncoding messageVersion="Soap11WSAddressing10" />
            <httpsTransport requireClientCertificate="true" />
          </binding>
        </customBinding>
      </bindings>
      <client>
        <endpoint address="http://testservicios.pge.red.uy/timestamp"
          behaviorConfiguration="PGEBehaviour" binding="customBinding"
          bindingConfiguration="CustomBinding_TimestampWSDLPortType"
          contract="Tutorial.TimestampService" name="TimestampServiceImplPort" />
      </client>
    </system.serviceModel>
</configuration>
```

Figura 15: Archivo app.config

Invocación del Servicio

1. Modificar el código de la clase *Program* para que quede similar a la figura 16

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using AGESIC.PGE;
using ClienteTutorial.Tutorial;

namespace ClienteTutorial
{
    public class Program
    {
        [STAThread]
        static void Main(string[] args)
        {
            PGEContext<TimestampServiceClient> contexto =
              PGEContext<TimestampServiceClient>.CreatePGEContext(
              "{{colocarNombre}}",
              "OU=TEST_TUTORIAL,O=TEST_PE",
              "urn:tokensimple");

              GetTimestamp ts = new GetTimestamp();
              GetTimestampResponse resp = contexto.Client.GetTimestamp(ts);
              Console.WriteLine(resp.Timestamp.ToString());
              Console.WriteLine("Servicio consumido correctamente");
              Console.ReadLine();
        }
    }
}
```

Figura 16: Crear Cliente PGE

2. Ejecutar el cliente haciendo clic derecho en el proyecto ClienteTutorial → *Debug* → *Start new instance.*

> **Importante:** Antes de correr el ejemplo asegúrese que la hora del servidor se encuentra sincronizada con la hora actual (incluyendo segundos). Si la hora se encuentra adelantada ocurrirá un error en la ejecución.
>
> Usted puede sincronizar la hora con el servidor NTP de la Plataforma: ntp.pge.red.uy