



**Gobierno
de Canarias**

Consejería de Presidencia,
Justicia y Seguridad

Dirección General de
Telecomunicaciones
y Nuevas Tecnologías



Platino

Plataforma de Interoperabilidad del
Gobierno de Canarias

Interoperabilidad de los servicios telemáticos de la
Administración Pública de la CAC

Página 1 de 9

Consumo de Platino desde .Net con clases proxy

Rev.	Fecha	Descripción
0	23/07/2010	Redacción inicial
Documento		
Ubicación en eCarpeta		
Preparado por		Revisado por
D. Gral. de Telecomunicaciones y Nuevas Tecnologías		D. Gral. de Telecomunicaciones y Nuevas Tecnologías
Fecha: 23/07/2010		Fecha: 23/07/2010
		Aprobado por
		D. Gral. de Telecomunicaciones y Nuevas Tecnologías
		Fecha: 23/07/2010



**Gobierno
de Canarias**

Consejería de Presidencia,
Justicia y Seguridad
Dirección General de
Telecomunicaciones
y Nuevas Tecnologías



Platino
Plataforma de Interoperabilidad del
Gobierno de Canarias

Consumo de Platino desde .Net con clases proxy

Página 2 de 9

Índice de contenido

1 INTRODUCCION	3
2 REQUISITOS TÉCNICOS	4
3 DESARROLLO DE LA SOLUCIÓN	5

 Gobierno de Canarias Consejería de Presidencia, Justicia y Seguridad Dirección General de Telecomunicaciones y Nuevas Tecnologías	 Platino Plataforma de Interoperabilidad del Gobierno de Canarias
Consumo de Platino desde .Net con clases proxy	Página 3 de 9

1 INTRODUCCION

Platino ha sido desarrollado utilizando los estándares abiertos más utilizados, aún así, el consumo de Platino desde aplicaciones desarrolladas sobre la plataforma .Net requiere de la adopción de soluciones a determinados problemas de comunicación.

Este documento refleja la propuesta de solución desarrollada por Galileo Ingeniería y Servicios S.A. para la aplicación UNIFICA de la Dirección General de Gasto y Presupuesto Publico de la Consejería de Hacienda del Gobierno de Canarias.

La solución se basa en la modificación del comportamiento del Web Services Enhancements (WSE) 3.0 en la construcción de las cabeceras de seguridad que son enviadas a Platino. Para modificar este comportamiento se genera una clase proxy en la que se sobrescribe el comportamiento de determinados métodos. La solución puede ser encapsulada en una dll que luego se utilizará en las aplicaciones desarrolladas o se pueden incorporar a la propia aplicación las clases que se describen en los siguientes apartados.

Otra de las modificaciones necesarias es la adaptación de la librería CAPICOM para que permita la firma con PKCS7 en utf-8. La versión original de la librería sólo permite firmar en utf-16 por lo que es necesario modificarla para dar soporte a la firma en utf-8 utilizada en Platino. Esta limitación y su solución se describe en el siguiente enlace.

<http://www.codeproject.com/KB/security/CapicomUTF8.aspx>

2 REQUISITOS TÉCNICOS

El desarrollo de la solución se basa en la modificación del comportamiento del Web Services Enhancements (WSE) del Microsoft .NET Framework 2.0. Actualmente no es posible aplicar la solución sobre el Microsoft .NET Framework 3.0 por las limitaciones para la parametrización del Windows Communication Foundation (WCF).

El desarrollo y consumo de la solución tiene los siguientes requisitos técnicos.

	Componente	Versión
Lenguaje de Programación	Vb, .Net, C#	
Frameworks	Microsoft .NET Framework 2.0	2.0
Librerías	Web Services Enhancements (WSE)	3.0
Herramienta de desarrollo	Microsoft Visual Studio 2005	8.0

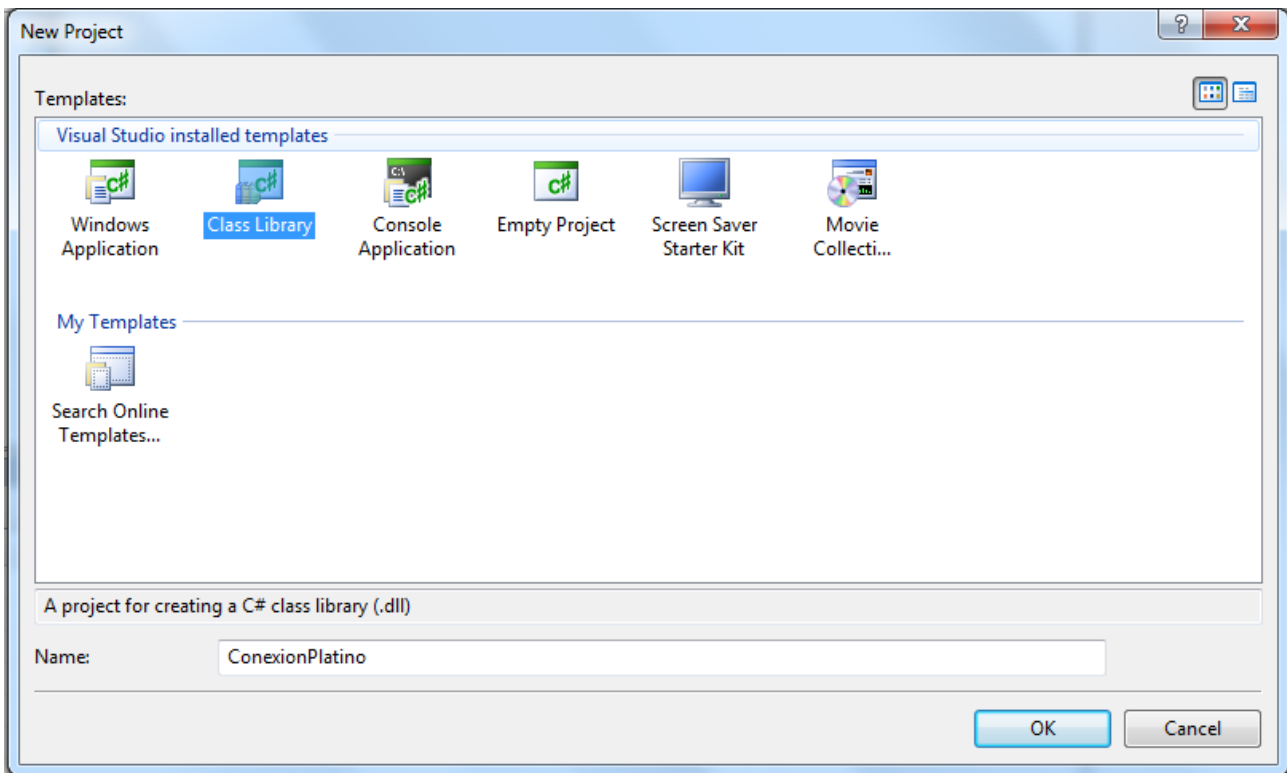
Las aplicaciones que deseen utilizar esta solución deben instalar Microsoft .NET Framework 2.0. Esta instalación es totalmente compatible con la instalación del Microsoft .NET Framework 3.0.



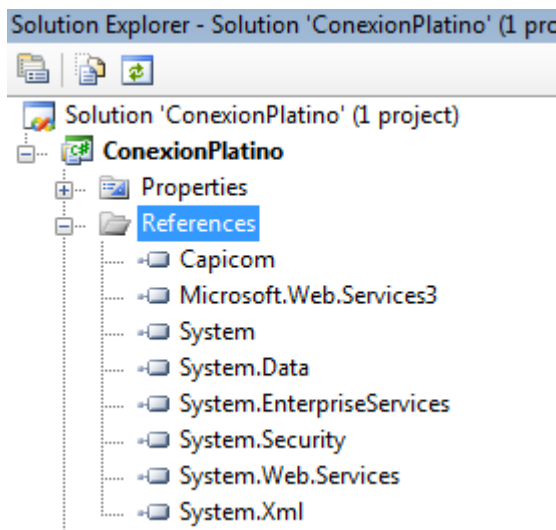
3 DESARROLLO DE LA SOLUCIÓN

Para el desarrollo de la solución se deben seguir los siguientes pasos.

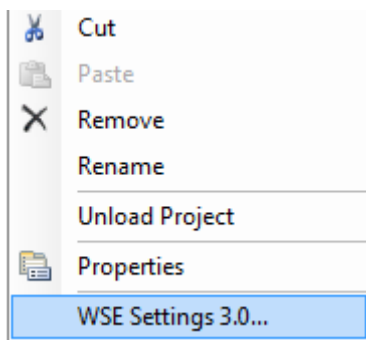
1. Crear un proyecto de tipo Librería de Clases



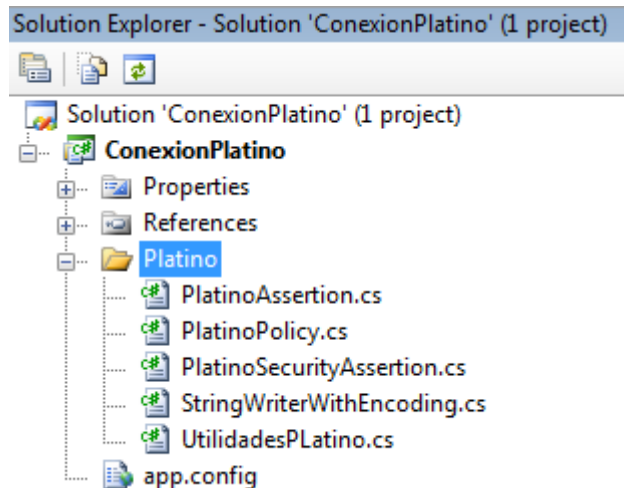
2. Debemos añadir las referencias a las librerías que serán necesarias para la generación del proyecto. Se necesitarán las librerías Microsoft.Web.Service3, System.Security y CAPICOM.



3. Configuramos el proyecto para utilizar Web Service Enhancements.

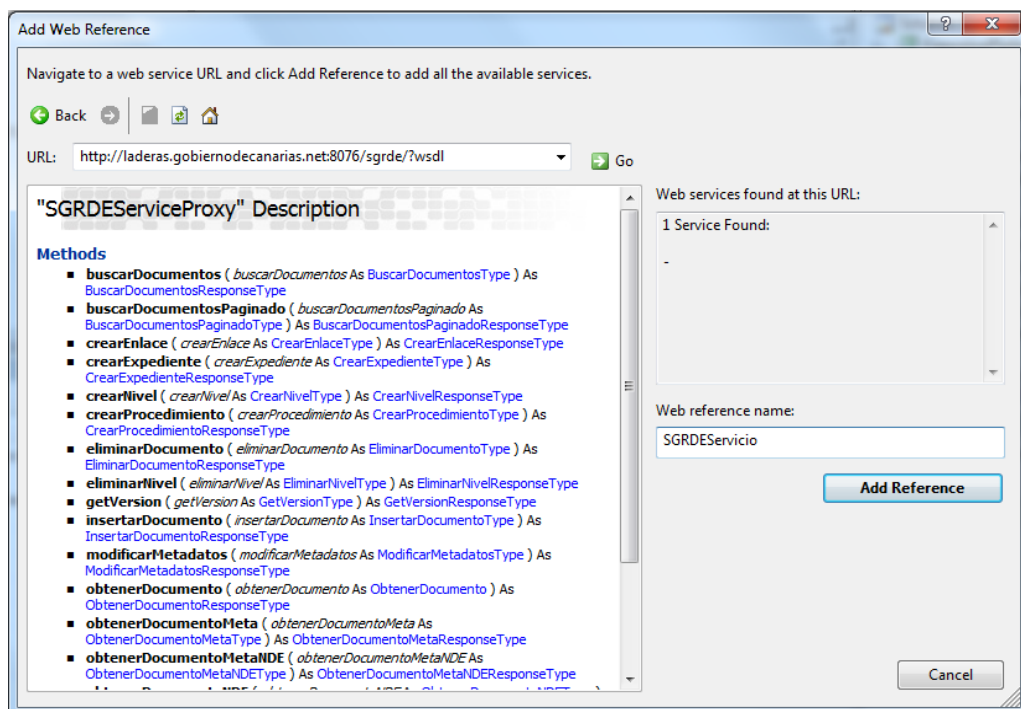


4. Creamos una carpeta para contener las clases que utilizaremos (ej: Platino). Dentro de esta carpeta debemos copiar las siguientes clases *PlatinoSecurityAssertion*, *PlatinoAssertion* y *PlatinoPolicy*. Las clases *UtilidadesPlatino* y *StringWriterWithEncoding* serán utilizadas para la firma.



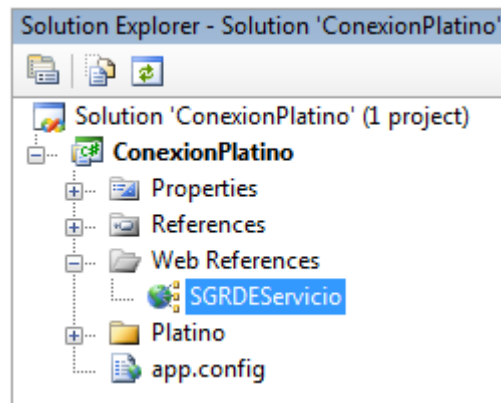
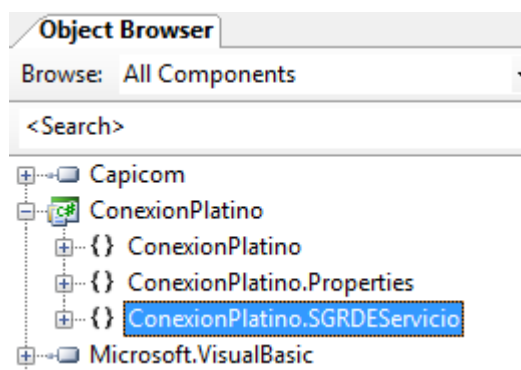
5. A continuación añadimos las Web Reference de los servicios de Platino que se vayan a utilizar.

Es necesario añadir una por cada servicio a utilizar. En este punto cabe destacar que será necesario generar clases proxy para cada entorno (pre-explotación y explotación) puesto que las URL de un mismo servicio varía entre entornos. En el ejemplo se muestra la creación del Web Reference para el Servicio de Repositorio de Documentos Electrónicos en el entorno de pre-explotación.





6. Al añadir el Web Reference se crean las clases proxy para el servicio.



Se generarán dos clases proxy, utilizaremos la que contiene el sufijo “Wse” que se genera al haber configurado el proyecto para usar el WSE como se indicaba en el punto 3.



A continuación se muestra parte del código de la clase generada automáticamente.

```

namespace ConexionPlatino.SGRDEServicio {
    using System.Diagnostics;
    using System.Web.Services;
    using System.ComponentModel;
    using System.Web.Services.Protocols;
    using System;
    using System.Xml.Serialization;
    |

    /// <remarks/>
    [System.CodeDom.Compiler.GeneratedCodeAttribute("System.Web.Services", "2.0.50727.42")]
    [System.Diagnostics.DebuggerStepThroughAttribute()]
    [System.ComponentModel.DesignerCategoryAttribute("code")]
    [System.Web.Services.WebServiceBindingAttribute(Name="SGRDEServiceProxyBinding", Namespace="http://platino.gob.es/SGRDEServicio", ConfigurationName="SGRDEServiceProxyBinding")]
    public partial class SGRDEServiceProxyWse : Microsoft.Web.Services3.WebServicesClientProtocol {

        private System.Threading.SendOrPostCallback crearNivelOperationCompleted;

        private System.Threading.SendOrPostCallback eliminarNivelOperationCompleted;
    }
}

```



7. El siguiente paso es crear una clase que derive de la clase proxy creada y que sobrescriba los métodos de creación y *WebResponse*. El código de esta clase se muestra a continuación:

```
using System;
using System.Collections.Generic;
using System.Text;
using ConexionPlatino.SGRDEServicio;
using System.Net;
using Microsoft.Web.Services3.Design;
using Microsoft.Web.Services3.Security;

namespace ConexionPlatino
{
    public class PlatinoSGRDEServiceProxyWse : SGRDEServiceProxyWse
    {
        public PlatinoSGRDEServiceProxyWse(string sUserSerialNumber, string sUsernameToken, string sPasswordUsernameToken, string[] nombre, string[] valores)
        {
            PlatinoSecurityAssertion customSecurity = new PlatinoSecurityAssertion(sUserSerialNumber, sUsernameToken, sPasswordUsernameToken, SignatureOptions.IncludeNone, nombre, valores);
            Policy policy = new Policy();
            policy.Assertions.Add(customSecurity);
            policy.Assertions.Add(new PlatinoAssertion());

            this.SoapVersion = System.Web.Services.Protocols.SoapProtocolVersion.Soap11;
            this.RequireMtom = true;
            // Aplicar policy
            this.SetPolicy(policy);
        }

        protected override WebResponse GetWebResponse(WebRequest request)
        {
            WebResponse response = base.GetWebResponse(request);
            // Si la respuesta es no MTOM como un error de SOAP, informe a la representación
            // que realmente no necesitan MTOM
            if (response.Headers[HttpResponseHeader.ContentType].ToLower().StartsWith("text/xml"))
            {
                this.RequireMtom = false;
            }
            return response;
        }
    }
}
```

Una vez generadas estas clases se podrán invocar las llamadas correspondientes desde la aplicación.

```
PlatinoSGRDEServiceProxyWse p = new PlatinoSGRDEServiceProxyWse(sUserSerialNumber,
sUsernameToken, sPasswordUsernameToken, null,null);
.....
.....
res = p.insertarDocumento(documento);
.....
```