

 <p>Gobierno de Canarias</p> <p>Consejería de Presidencia, Justicia y Seguridad</p> <p>Dirección General de Telecomunicaciones y Nuevas Tecnologías</p>	 <p>Platino</p> <p>Plataforma de Interoperabilidad del Gobierno de Canarias</p>
<p>Interoperabilidad de los servicios telemáticos de la Administración Pública de la CAC</p>	<p>Página 1 de 17</p>

Proxy .Net/J2EE

Rev.	Fecha	Descripción
0	04/03/2010	Redacción inicial del documento
Documento :		IIIC_2010-04-03_ATLAS-ProryNetJava.
Ubicación en eRoom:		
Preparado por		Revisado por
Cabildo Insular de Tenerife		
Fecha: 04/03/2010		Fecha:
		Fecha:

 <p>Gobierno de Canarias Consejería de Presidencia, Justicia y Seguridad Dirección General de Telecomunicaciones y Nuevas Tecnologías</p>	 <p>Platino Plataforma de Interoperabilidad del Gobierno de Canarias</p>
Proxy .Net/J2EE	Página 2 de 17

ÍNDICE

1 INTRODUCCIÓN.....3

2 REQUISITOS TÉCNICOS.....4

 2.1 CLIENTE .NET.....4

 2.2 Proxy PLATINO.....4

3 PROCESO DE DESARROLLO DEL PROXY.....6

4 PROCESO DE DESARROLLO DEL CLIENTE .NET.....12

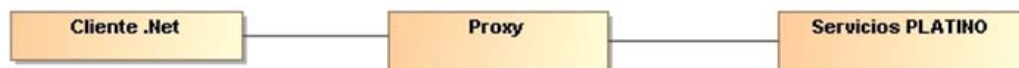
 <p>Gobierno de Canarias Consejería de Presidencia, Justicia y Seguridad Dirección General de Telecomunicaciones y Nuevas Tecnologías</p>	 <p>Platino Plataforma de Interoperabilidad del Gobierno de Canarias</p>
Proxy .Net/J2EE	Página 3 de 17

1 INTRODUCCIÓN

En este documento se describe la solución de carácter provisional que se ha empleado en el Instituto Insular de Informática y Comunicaciones del Cabildo de Tenerife, en adelante IIIC, para el consumo de los servicios de PLATINO desde la plataforma .Net.

Esta solución se plantea debido a que se han encontrado dificultades técnicas para la generación de las cabeceras de seguridad desde .Net y en el uso de MTOM desde la plataforma. En la solución aquí propuesta se resuelve el problema para algunos métodos del servicio de Gestión de Repositorio de Documentos Electrónicos, en adelante SGRDE, aunque se puede emplear la misma estrategia para el resto de servicios de PLATINO.

Se ha implementado un proxy en Java que se encarga de realizar las invocaciones sobre la plataforma y a su vez se genera un cliente .Net al proxy implementado.



 <p>Gobierno de Canarias Consejería de Presidencia, Justicia y Seguridad Dirección General de Telecomunicaciones y Nuevas Tecnologías</p>	 <p>Platino Plataforma de Interoperabilidad del Gobierno de Canarias</p>
Proxy .Net/J2EE	Página 4 de 17

2 REQUISITOS TÉCNICOS

2.1 CLIENTE .NET

	Componente	Versión	Observaciones
Lenguaje de programación.	Vb .Net, C#	2.0	
Entorno de ejecución (servidor)	IIS Microsoft Windows	5.1 o superior 2000 Server o superior	
Estándares Frameworks APIS	.Net	3.0	Necesaria para la correcta ejecución de las librería WCF.
Librería de terceros	WCF	3.0	Necesaria para la recepción de ficheros adjuntos a través de mensajes SOAP (MTOM) http://msdn.microsoft.com/es-es/library/ms735119.aspx
Herramientas de desarrollo	Microsoft Visual Studio 2005	8.0	

2.2 Proxy PLATINO

	Componente	Versión	Observaciones
Lenguaje de programación.	Java	1,52.0	
Entorno de ejecución (servidor)	Apache Tomcat	5.0 o superior	Puede servir cualquier otro servidor de aplicaciones compatible con la especificación J2EE 1.2.
Estándares Frameworks APIS	Spring Apache CXF	2.0 2.0.7	



**Gobierno
de Canarias**

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



Platino
Plataforma de Interoperabilidad del
Gobierno de Canarias

Proxy .Net/J2EE

Página 5 de 17

Librería de terceros	PlatinoWSSInterceptor-2.0.jar	2.0	Librería utilizada para establecer las cabeceras de seguridad de PLATINO. Para más información consultar http://www.gobiernodecanarias.org/platino/formacion.html
Herramientas de desarrollo	Eclipse	3.4.2	

 <p>Gobierno de Canarias Consejería de Presidencia, Justicia y Seguridad Dirección General de Telecomunicaciones y Nuevas Tecnologías</p>	 <p>Platino Plataforma de Interoperabilidad del Gobierno de Canarias</p>
Proxy .Net/J2EE	Página 6 de 17

3 PROCESO DE DESARROLLO DEL PROXY

En este apartado se explica cómo se ha creado el proxy. Los mismos pasos se pueden seguir para crear un proxy similar pero que de cobertura a otros métodos o servicios de PLATINO.

- Se crea un proyecto Web con la ayuda de Maven.

Por ejemplo:

```
mvn archetype:create -DgroupId=your.groupId
-DarchetypeArtifactId=maven-archetype-webapp
-DartifactId=Proxy
```

- Se prepara el proyecto para ser importando en Eclipse. Desde el directorio del proyecto, se ejecuta

```
mvn eclipse:eclipse
```

- Se abre el Eclipse y se importa el proyecto. File -> Import... y se selecciona el fichero .project que se acaba de generar con el maven.

- Se genera el cliente de los servicios web de PLATINO, empleando Apache CXF. Este producto es un framework de servicios open source que puede descargarse desde la siguiente URL:

<http://cxf.apache.org/download.html>

Para generar un cliente únicamente se necesita conocer la URL del WSDL del servicio web que se va a consumir y el path del sistema de ficheros local donde se quiere generar las clases clientes.

El comando que se debe ejecutar es el siguiente:

```
wsdl2java -d <output-directory> wsdlfile
```

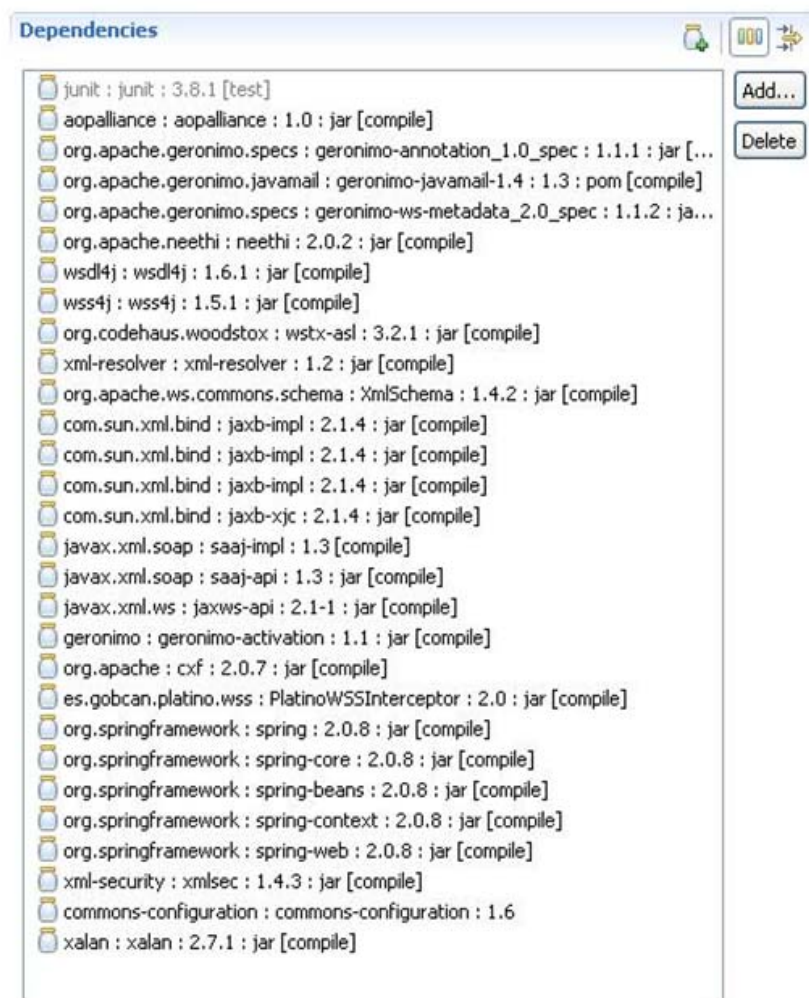
El ejecutable 'wsdl2java' se encuentra en el directorio 'bin' de la distribución de CXF.

A continuación se indica un ejemplo de invocación:

```
> wsdl2java -d c:\workspace\proxy\src
http://laderas.gobiernodecanarias.net:8076/sgrde/?wsdl
```

Tras la ejecución, el generador CXF habrá creado todos los ficheros java necesarios para la invocación de los métodos del Servicio Web.

- Se deben añadir las dependencias necesarias en el fichero pom.xml del proyecto.



Se crean los servicios que necesitamos invocar desde .Net, para ello es necesario crear una interfaz con la declaración de los servicios que se van a utilizar.



Ejemplo:

```
package es.i3c.proxy;

import javax.jws.WebParam;
import javax.jws.WebResult;
import javax.jws.WebService;
import es.gobcan.platino.servicios.sgrde.Documento;
import es.gobcan.platino.servicios.sgrde.DocumentoBase;
import es.gobcan.platino.servicios.sgrde.ElementoNoEncontradoException;
import es.gobcan.platino.servicios.sgrde.ErrorInternoException;
import es.gobcan.platino.servicios.sgrde.ObtenerDocumentoResponseType;

@WebService
public interface ProxyPLATINO {

    String getVersionSGRDE();

    @WebResult(name="ObtenerDocumentoResponse")
    ObtenerDocumentoResponseType
    obtenerDocumento(@WebParam(name="uri") String uri) throws
    ElementoNoEncontradoException, ErrorInternoException;

    byte[] obtenerDocumentoByte(@WebParam(name="uri") String
    uri) throws ElementoNoEncontradoException, ErrorInternoException;

    Documento obtenerDocumentoMeta(@WebParam(name="uri") String
    uri) throws ElementoNoEncontradoException, ErrorInternoException;

    DocumentoBase obtenerMetaDoc(@WebParam(name="uri") String
    uri) throws ElementoNoEncontradoException, ErrorInternoException;

}
```

- Se crea una clase que implemente a la interfaz.



Ejemplo:

```
@WebService(endpointInterface = "es.i3c.proxy.ProxyPLATINO")
public class ProxyPLATINOImpl implements ProxyPLATINO {

    public String getVersionSGRDE() {

        URL wsd1 = null;
        try {
            wsd1 = new URL(Config.getString("SGRDE.wsd1"));
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
        QName SERVICE = new QName(Config.getString("SGRDE.Service"),
Config
            .getString("SGRDE.ServiceProxy"));

        SGRDEServiceProxy sgrde = new SGRDEServiceProxy(wsd1,
SERVICE);

        SGRDEServicePortType port =
sgrde.getSGRDEServiceProxyPort();

        // pruebas
        Client client = ClientProxy.getClient(port);
        HTTPConduit http = (HTTPConduit) client.getConduit();

        HTTPClientPolicy httpClientPolicy = new HTTPClientPolicy();

        httpClientPolicy.setConnectionTimeout(36000);
        httpClientPolicy.setAllowChunking(false);
        httpClientPolicy.setMaxRetransmits(0);

        http.setClient(httpClientPolicy);

        // Establecemos las Cabeceras de Seguridad
        // inclusion de cabecera de seguridad para webservice
        PlatinoCXFSecurityHeaders
            .addSoapWSHeader(
                PORT,
                PlatinoCXFSecurityHeaders.SOAP_11,
                Config.getString("SGRDE.username"),
                Config
                .getString("org.apache.ws.security.crypto.merlin.keystore.alias"),
                ClientPasswordHandler.class.getName(), null);

        return port.getVersion();
    }
}
```

 <p>Gobierno de Canarias</p> <p>Consejería de Presidencia, Justicia y Seguridad</p> <p>Dirección General de Telecomunicaciones y Nuevas Tecnologías</p>	 <p>Platino Plataforma de Interoperabilidad del Gobierno de Canarias</p>
Proxy .Net/J2EE	Página 10 de 17

La inclusión de las cabeceras de seguridad de PLATINO, se encuentran detalladas en el documento Acceso a Servicios en pre-explotación en el menú de formación de la Web del Gobierno de Canarias.

<http://www.gobiernodecanarias.org/platino/formacionPRE.html>.

Se muestra un ejemplo de la configuración de CXF del proyecto.

Ejemplo de fichero cxf.xml:

```
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:jaxws="http://cxf.apache.org/jaxws"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
http://cxf.apache.org/jaxws
http://cxf.apache.org/schemas/jaxws.xsd">

<import resource="classpath:META-INF/cxf/cxf.xml"/>
<import resource="classpath:META-INF/cxf/cxf-extension-soap.xml"/>
<import resource="classpath:META-INF/cxf/cxf-servlet.xml"/>

<jaxws:endpoint id="proxyJustificantePDF"
implementor="es.13c.proxy.ProxyPLATINOImpl"
address="/proxyJustificantePDF">
<jaxws:properties>
<entry key="mtom-enabled" value="false"/>
</jaxws:properties>
</jaxws:endpoint>

</beans>
```



Ejemplo de configuración del fichero web.xml del proyecto:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>

  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>WEB-INF/cxf.xml</param-value>
  </context-param>

  <listener>
    <listener-class>
      org.springframework.web.context.ContextLoaderListener
    </listener-class>
  </listener>

  <!-- SERVLET DE CXF -->
  <servlet>
    <servlet-name>CXFServlet</servlet-name>
    <display-name>ProxyPlatino</display-name>
    <servlet-class>
      org.apache.cxf.transport.servlet.CXFServlet
    </servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>CXFServlet</servlet-name>
    <url-pattern>/services/*</url-pattern>
  </servlet-mapping>

</web-app>
```

Una vez implementada la clase se compila el proyecto y se empaqueta. Para realizar esta tarea se ejecuta desde la consola en el directorio raíz:

```
mvn install
```

Si todo ha sido correcto se obtiene el war de la aplicación, preparado para desplegar en un servidor J2EE.

 <p>Gobierno de Canarias Consejería de Presidencia, Justicia y Seguridad Dirección General de Telecomunicaciones y Nuevas Tecnologías</p>	 <p>Platino Plataforma de Interoperabilidad del Gobierno de Canarias</p>
Proxy .Net/J2EE	Página 12 de 17

4 PROCESO DE DESARROLLO DEL CLIENTE .NET

Para generar un cliente del Proxy en .Net se utiliza WCF. Se puede obtener información de WCF en <http://msdn.microsoft.com/es-es/library/ms735119.aspx>.

Para generar la clase que implementa el cliente del Proxy, se utiliza [ServiceModel Metadata Utility Tool \(Svcutil.exe\)](#).

Se abre una consola en Windows y se ejecuta lo siguiente:

```
svcutil.exe /language:vb /out:generatedProxy.vb /config:app.config http://ruta_donde_tengamos_desplegado_el_proxy?wsdl
```

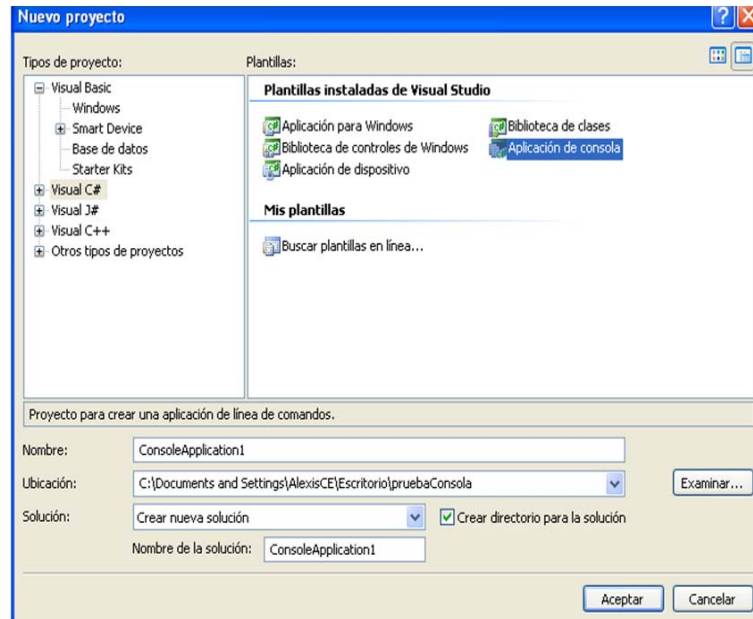
donde:

/language: puede tener los valores de vb si queremos el código en Visual Basic.net o cs si lo queremos en C#

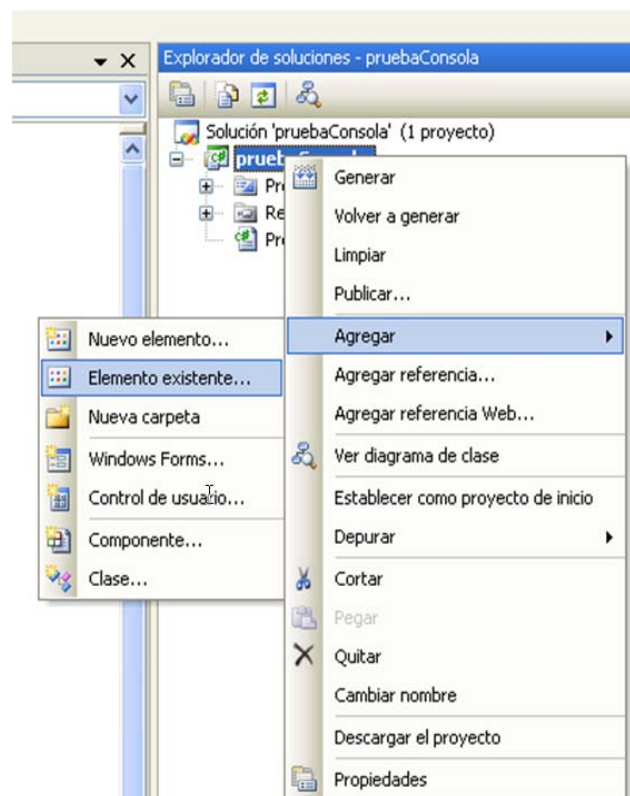
/out: es el nombre que queremos darle a la clase generada.

/config: nombre del fichero de configuración. app.config en caso de aplicación de consola y web.config en caso de aplicación web.

Se inicial el Visual Studio y se crea una aplicación de consola de ejemplo.



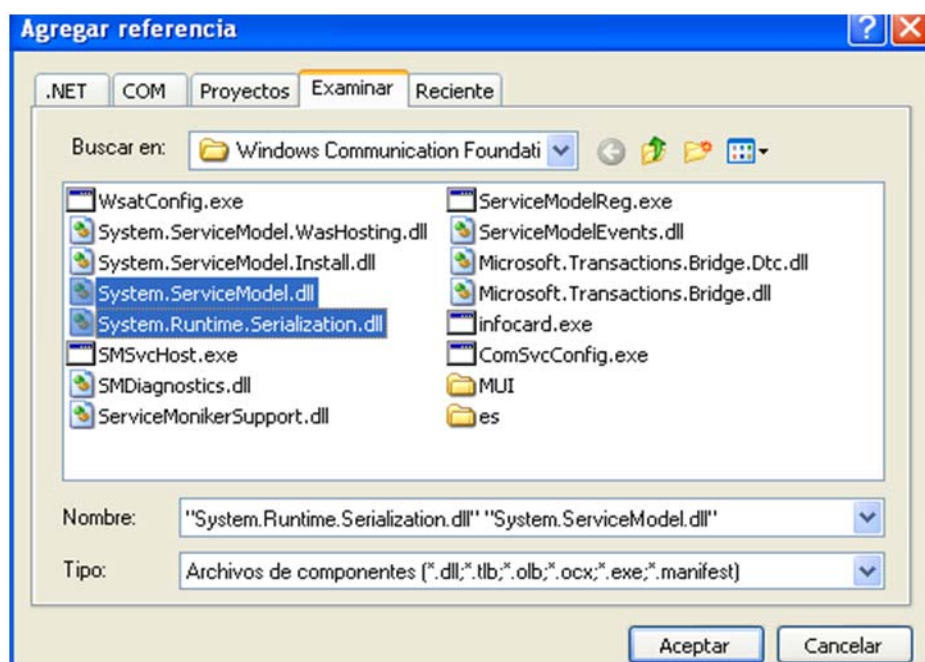
Se agrega al proyecto la clase generada con svcutil.exe y el fichero de configuración.



 <p>Gobierno de Canarias Consejería de Presidencia, Justicia y Seguridad Dirección General de Telecomunicaciones y Nuevas Tecnologías</p>	 <p>Platino Plataforma de Interoperabilidad del Gobierno de Canarias</p>
Proxy .Net/J2EE	Página 14 de 17

Agregue una referencia a **System.ServiceModel.dll** y **System.Runtime.Serialization** para el proyecto:

1. Haga clic con el botón secundario en la carpeta **Referencias** del proyecto **Cliente** en el **Explorador de soluciones** y seleccione **Agregar referencia**.
2. Seleccione la ficha **Examinar**, navegue hasta **C:\Windows\Microsoft.NET\Framework\v3.0\Windows Communication Foundation** y seleccione el ensamblado desde allí.



Se implementa la invocación al proxy, por ejemplo:



**Gobierno
de Canarias**

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



Platino
Plataforma de Interoperabilidad del
Gobierno de Canarias

Proxy .Net/J2EE

Página 15 de 17

```
static void Main(string[] args)
{
    Console.WriteLine("Prueba de invocación a PLATINO desde
.Net");

    ProxyPLATINOCient proxy = new ProxyPLATINOCient();

    //Invocacion GetVersion
    Console.WriteLine ("Version de SGRDE: " +
proxy.getVersionSGRDE());

    Console.WriteLine("Introduzca la uri de un documento para
obtener su meta-información: ");
    String uri = "";

    uri = Console.ReadLine();

    //Invocación obtenerMetaDoc
    if (uri != null){
        try
        {
            DocumentoBase doc = proxy.obtenerMetaDoc(uri);
            Console.WriteLine("Tipo-mime del documento:" +
doc.Tipo_Mime);
        }
        catch (Exception e)
        {
            Console.WriteLine("Error:" + e.Message );
        }
    }

    Console.WriteLine("Presione Enter para Salir...");
    Console.ReadLine();
}
```



Fichero de configuración de la aplicación .Net de ejemplo.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="ProxyPLATINOImplServiceSoapBinding"
closeTimeout="00:01:00"
openTimeout="00:01:00" receiveTimeout="00:10:00"
sendTimeout="00:01:00"
allowCookies="false" bypassProxyOnLocal="false"
hostNameComparisonMode="StrongWildcard"
maxBufferSize="65536" maxBufferPoolSize="524288"
maxReceivedMessageSize="65536"
messageEncoding="Text" textEncoding="utf-8"
transferMode="Buffered"
useDefaultWebProxy="true">
          <readerQuotas maxDepth="32"
maxStringLength="8192" maxArrayLength="16384"
maxBytesPerRead="4096"
maxNameTableCharCount="16384" />
          <security mode="None">
            <transport clientCredentialType="None"
proxyCredentialType="None"
realm="" />
            <message clientCredentialType="UserName"
algorithmSuite="Default" />
          </security>
        </binding>
      </basicHttpBinding>
    </bindings>
    <client>
      <endpoint
address="http://localhost:8080/proxyJustificantePDF/services/proxyJustif
icantePDF"
binding="basicHttpBinding"
bindingConfiguration="ProxyPLATINOImplServiceSoapBinding"
contract="ProxyPLATINO" name="ProxyPLATINOImplPort" />
    </client>
  </system.serviceModel>
</configuration>
```

Se observa que el tipo de enlace es **basicHttpBinding** en el cual no hay seguridad definida, en caso de que se establezca seguridad entre el Proxy y la aplicación Cliente .Net se pueden utilizar otras configuraciones de enlace.



La siguiente tabla muestra las características de cada uno de los enlaces proporcionados por el sistema

Enlace	Interoperabilidad	Seguridad (valor predeterminado)	Codificación (valor predeterminado)	Transmisión por secuencias (Predeterminado).
BasicHttpBinding	Basic Profile 1.1	(Ninguno), transporte, mensaje, mixto	Texto, (MTOM)	Sí (almacenado en búfer)
WSHttpBinding	WS	Transporte, (mensaje), mixto	(Texto), MTOM	No
WSDualHttpBinding	WS	(Mensaje), ninguno	(Texto), MTOM	No
WSFederationHttpBinding	WS-Federation	(Mensaje), mixto, ninguno	(Texto), MTOM	No
NetTcpBinding	.NET	(Transporte), mensaje, ninguno, mixto	Binario	Sí (almacenado en búfer)
NetNamedPipeBinding	.NET	(Transporte), ninguno	Binario	Sí (almacenado en búfer)
NetMsmqBinding	.NET	Mensaje, (Transporte), Ninguno	Binario	No
NetPeerTcpBinding	Del mismo nivel	(Transporte)		No
MsmqIntegrationBinding	MSMQ	(Transporte)	n/a	No
BasicHttpContextBinding	Basic Profile 1.1	(Ninguno), transporte, mensaje, mixto	Texto, (MTOM)	Sí (almacenado en búfer)
NetTcpContextBinding	.NET	(Transporte), mensaje, ninguno, mixto	Binario	Sí (almacenado en búfer)
WSHttpContextBinding	WS	Transporte, (mensaje), mixto	Texto, (MTOM)	No

Para ampliar la información <http://msdn.microsoft.com/es-es/library/ms730879.aspx>