Manual de instalación, configuración e integración Cl@ve para Proveedores de Servicio para .NET

Resumen: Esta es un manual para la instalación, configuración e integración a la plataforma Cl@ve para proveedores de servicio. También integradores de sistemas se pueden beneficiar de este manual. Este manual debe ser leído por administradores e integradores responsables de integrar Cl@ve en una aplicación en tecnología .NET.

Historial del documento

Version	*Date*	*Modification reason*	*Modified by*
1.0	01/02/2012	Versión inicial	
2.0	11/06/2014	Versión ajustada a las nuevas tecnologías	Alfredo de Cea Belló
2.1	02/11/2015	Añadida configuración para nuevas funcionalidades	Alfredo de Cea Belló
2.2	17/12/2015	Actualización del parámetro “forceAuth”	Alfredo de Cea Belló

Índice

HISTORIAL DEL DOCUMENTO **3**
ÍNDICE **4**
LISTA DE ABREVIATURAS **5**
RESUMEN EJECUTIVO **6**
1 INTRODUCCIÓN **7**
2 ANTES DE EMPEZAR **8**
2.1 KEYSTORE 8
2.1.1 INSTALANDO LOS CERTIFICADOS 8
2.2 CONFIGURACIÓN DEL SERVIDOR 9
2.2.1 ISS 7.5 9
3 INICIO RÁPIDO **10**
3.1 ABRA VISUAL STUDIO 2012 10
3.2 CONFIGURACIONES 10
3.3 COMPILACIÓN Y DESPLIEGUE RÁPIDO 10
3.3.1 DESPLIEGUE IIS 7.5 10
4 INSTALANDO DEMOSP.NET **12**
4.1 FICHEROS DE CONFIGURATION 12
4.1.1 WEB.CONFIG 12
5 INICIAR LA APLICACIÓN **13**
6 SINGLE SIGN ON **16**
6.1 INTEGRACIÓN DE LA DESCONEXIÓN 16
6.2 FORZAR LA AUTENTICACIÓN 16
7 SAML ENGINE API **17**
7.1 EJEMPLOS DE USO 17
7.1.1 GENERANDO UNA PETICIÓN DE AUTENTICACIÓN AUTOMÁTICA 17
7.1.2 GENERANDO UNA PETICIÓN DE AUTENTICACIÓN A MEDIDA 17
7.1.3 VALIDANDO Y LEYENDO UNA RESPUESTA DE AUTENTICACIÓN 17
8 ANEXO: PETICIÓN DE ALTA COMO PROVEEDOR DE SERVICIOS CON ACCESO A LA PLATAFORMA STORK **19**

Lista de abreviaturas

<Abreviatura>	<Explicación>
SP	Service Provider (Proveedor de Servicio)
IdP	Identity Provider (Proveedor de Identidad)
VS 2012	Visual Studio 2012

Resumen ejecutivo

Este documento ofrece información detallada sobre como configurar, crear y desplegar en .NET una aplicación para un Proveedor de Servicios (SP) para su uso en la plataforma Cl@ve.

Como es necesaria la existencia de un IIS 7.5 para desplegar la aplicación SP, el documento comienza dando una información básica sobre el servidor.

Después de eso, se describe qué necesita saber el usuario sobre las posibles configuraciones para su proyecto.

Tras leer este manual, el administrador o integrador debería ser capaz de configurar, crear y desplegar una aplicación que sea capaz de conectarse a la plataforma Cl@ve.

1 Introducción

Este documento está dividido en varios capítulos con el fin de permitir al lector acceder fácilmente a las secciones más relevantes para el escenario específico en el que esté trabajando.

Primero, se aborda la configuración del servidor IIS 7.5 (preparándolo para el despliegue de la aplicación SP) y se explica cómo instalar los certificados necesarios.

A continuación, en otro punto se describe cómo poner en marcha la aplicación en unos pocos minutos.

Después de eso se entra en más detalle en las posibles configuraciones, para que el lector comprenda las posibilidades que le ofrece el sistema y modifique la configuración básica adaptándola a sus propias necesidades.

2 Antes de empezar

Asegúrese de que dispone de un Servidor Windows con IIS 7.5 (viene por defecto instalado con Windows 7) instalado con .NET Framework 4.5, ya que la aplicación fue desarrollada con Visual Studio Project. También necesitará VS 2012 y también .NET Framework 4.5 instalado en su máquina local (para modificar y crear el paquete SP).

En la siguiente sección de dan las instrucciones para configurar el certificado que utilizará el motor SAML Engine para firmar los token antes de desplegar el proyecto en IIS 7.5.

Con el objeto de facilitar la lectura del documento se utilizará la siguiente variable:

`$SP_PACKAGE_DIRECTORY – El directorio base con el ZIP con el contenido del paquete SP.`

2.1 Keystore

La aplicación SP usa el Windows Certstore / keystore y, por tanto, el certificado con el que firmar el SAML y el certificado del PEPS español con el que conectar debe estar instalado en él.

2.1.1 Instalando los Certificados

Para instalar el certificado para firmar el SAML y el certificado de la pasarela, necesita seguir los siguientes pasos:

1. Pulse el menú "Inicio" y abra la consola Windows Certificate Management ejecutando el comando "mmc.exe";
2. Presione CRTL + M y en la ventana que aparece seleccione "Certificados" y pulse "Agregar";
3. Pulse en el checkbox "Cuenta de equipo", luego pulse "Siguiente" y finalmente seleccione la opción "Equipo Local") y pulse "Finalizar";
4. Cierre la ventana pulsando "Aceptar";
5. Expanda el directorio "Personal" en el menú "Certificados (Equipo Local)";
6. Pulse el botón derecho en la carpeta "Certificados", seleccione "Todas las tareas" y entonces pulse en la opción "Importar…";
7. Siga el wizard para instalar el certificado de la aplicación SP (certificado P12 ó PFX);
8. Expanda la carpeta "Personal" en el menú "Certificados (Equipo Local)" y abra el certificado instalado haciendo doble clic sobre él.
9. Apunte el valor del campo "Huella digital" (sin espacios entre medias) de la pestaña "Detalles", ya que será necesario más adelante.

10. Expanda la carpeta "Entidades de certificación raíz de confianza" y pulse el botón derecho sobre la carpeta "Certificados", seleccione "Todas las tareas" y entonces pulse la opción "importar…";
11. Siga el wizard para instalar el certificado de la pasarela (certificado CER).

Si no obtiene ningún error, acaba de instalar el certificado para el motor SAML Engine y el certificado de la pasarela.

2.2 Configuración del Servidor

2.2.1 ISS 7.5

Antes de desplegar la aplicación SP, necesita configurar un servidor IIS 7.5:

1. Abra el explorador panel de control y vaya a Programas->Activar o desactivar las características de Windows.
2. Active la casilla de "Internet Information Services" (tiene que estar totalmente marcada, de forma que es posible que sea necesario abrir el desplegable y activar una a una todas las características).
3. Abra el Administrador del servidor a través del menú Inicio->Administrador de Internet Information Services (IIS). A través de esta herramienta podrá iniciar y detener el servidor, así como los pools de aplicaciones (Grupos de Aplicaciones).
4. Compruebe si en el apartado IIS->Certificados de Servidor aparece el certificado que ha instalado previamente. En caso de que no sea así impórtelo pulsando en "Importar…"

3 Inicio Rápido

Este inicio rápido le permitirá configurar, compilar y ejecutar el proyecto en pocos minutos, pero antes de empezar necesita configurar su servidor tal y como se describe en el capítulo anterior (vea "Antes de empezar"). Una información más detallada sobre ficheros de configuración y atributos la tiene disponible en el siguiente capítulo.

3.1 Abra Visual Studio 2012

La aplicación SP ha sido creada en Visual Studio 2012 y, por tanto, necesita tenerlo instalado (además de.NET Framework v4.5, que por cierto no es compatible con versiones anteriores de Visual Studio).

Una vez que haya instalado VS 2012, necesita importar el proyecto:

1. Abra el directorio `$SP_PACKAGE_DIRECTORY\DemoSP.NET (proyecto VS2012)`.
2. Haga doble click sobre el fichero "`DemoSP.NET.sln`".
3. (El proyecto será abierto por VS 2012).

3.2 Configuraciones

En VS2012, sobre el proyecto DemoSP.NET, edite el fichero `web.config` y modifique las siguientes propiedades:

- **SPReturnURL** =http(s):// **insert.your.ip.here** /SP/HandleResponse.aspx
- **SPLogoutReturnURL** =http(s):// **insert.your.ip.here** /SP/HandleLogoutResponse.aspx
- **ES.SPEPSURL** =http(s)://**Cl@ve.server.ip**/Proxy/ServiceProvider
- **SendTo** = http(s)://**Cl@ve.server.ip**/Proxy/ServiceProvider
- **LogoutSendTo** = http(s)://**Cl@ve.server.ip**/Proxy/LogoutAction
- **SamlCertificate = huella.digital.del.certificado.instalado.para.firmar.peticiones** (apuntada anteriormente)

SendTo será utilizado para enviar las peticiones automáticas, mientras que ES.SPEPSURL será usado para las peticiones a medida.

Ha concluido la configuración del SP y está preparado para ponerlo en marcha en su propia dirección IP.

3.3 Compilación y Despliegue rápido

3.3.1 Despliegue IIS 7.5

Debe generar la aplicación sobre VS 2012 antes de desplegarlo sobre el servidor ISS 7.5:

1. Abra VS 2012;

2. En "Explorador de soluciones", botón derecho sobre el proyecto "DemoSP.NET" y pulse "Propiedades".
3. En el apartado "Aplicación" asegúrese de que la aplicación usa la versión .Net Framework 4.5.1.
4. En el apartado "Web" seleccione "Usar servidor web de IIS local", en "URL del proyecto" escriba http://localhost/SP y a continuación pulse en "Crear directorio virtual" y guarde cambios
5. En "Explorador de soluciones", botón derecho sobre el proyecto "Common" y pulse "Generar";
6. En "Explorador de soluciones", botón derecho sobre el proyecto "Common.SAML" y pulse "Generar";
7. En "Explorador de soluciones", botón derecho sobre el proyecto "DemoSP.NET" y pulse "Generar";
8. Por último, abra su navegador y vaya a la siguiente página: "http(s)://**insert.your.ip.here** /SP/"

4 Instalando DemoSP.NET

4.1 Ficheros de configuration

El proyecto DemoSP.NET viene con una configuración que puede resultar insuficiente. En esta sección se explica cada propiedad.

4.1.1 web.config

El fichero `web.config` prove las principales configuraciones para el SP.

Key	Description
`SPProviderName`	Nombre del Proveedor de Servicio
`SPCountry`	País del Proveedor de Servicio (SP)
`SPQAALevel`	Nivel QAA del SP
`SPReturnURL`	URL usada cuando la pasarela finaliza el proceso (es la dirección de respuesta)
`SPLogoutReturnURL`	URL usada cuando la pasarela finaliza el proceso de Logout (es la dirección de respuesta)
`ES.SPEPSURL`	Es la URL de la pasarela (dirección a la que enviar las solicitudes a medida)
`SendTo`	Es la URL de la pasarela (dirección a la que enviar las solicitudes automáticas)
`LogoutSendTo`	Es la URL de Logout de la pasarela (dirección a la que enviar las solicitudes de logout)
`SPSector`	Sector del SP
`SPApplication`	Aplicación del SP

Nota que hay 2 configuraciones más, que deberían tomar como valor el nombre del proveedor de servicios.

Atributos disponibles para el SP:

Key	Description
AttributeSeparator	El separador de la lista de atributos
PersonalAttributeList	La lista de atributos personales separados con punto y coma
PersonalAttributeName.NS	El namespace del AttributeName personal
BusinessAttributeList	La lista de atributos de negocio separados con punto y coma
BusinessAttributeName.NS	El namespace del AttributeName de negocio
LegalAttributeList	La lista de atributos legales separados con punto y coma
LegalAttributeName.NS	El namespace del AttributeName legal
Adicionalmente existen tantas entradas como atributos. Cada entrada contiene el nombre del atributo y su traducción al español	

Configuraciones del SAML Engine:

Key	Description
`SamlCertificate`	La huella digital (**no el serial number!**) del certificado a obtener del keystore de la máquina local (sin los separadores);
`SamlValidTimeframe`	Número de minutos de validez para el token
`NSQAALevel`	El namespace del atributo SAML's QAA Level
`NSQAALevelPrefix`	El prefijo del nivel SAML's QAA
`NSReqAttrs`	El namespace de atributos para las SAML Requests
`NSReqAttrsPrefix`	El prefijo de atributos para las SAML Requests
`NSReqAttr`	El namespace de atributos para las SAML Requests
`NSReqAttrPrefix`	El prefijo de atributos para las SAML Requests

5 Iniciar la aplicación

Abra su navegador y vaya a http(s)://your.ip.addres/SP/ (si su servidor está escuchando en otro puerto cambie a URL consecuentemente). Debería ver una página similar a la Figura 1.



Figura 1: Página de Inicio - DemoSP:NET

Aquí puede mandar una petición automática (Mandar petición), que solicitará a la pasarela apuntada por "SendTo" una identificación en la que se piden los campos "eIdentifier", "givenName" y "surname" (Identificador electrónico, nombre y apellidos).

El otro botón (Petición a medida) lleva a la ventana mostrada en la Figura 2:

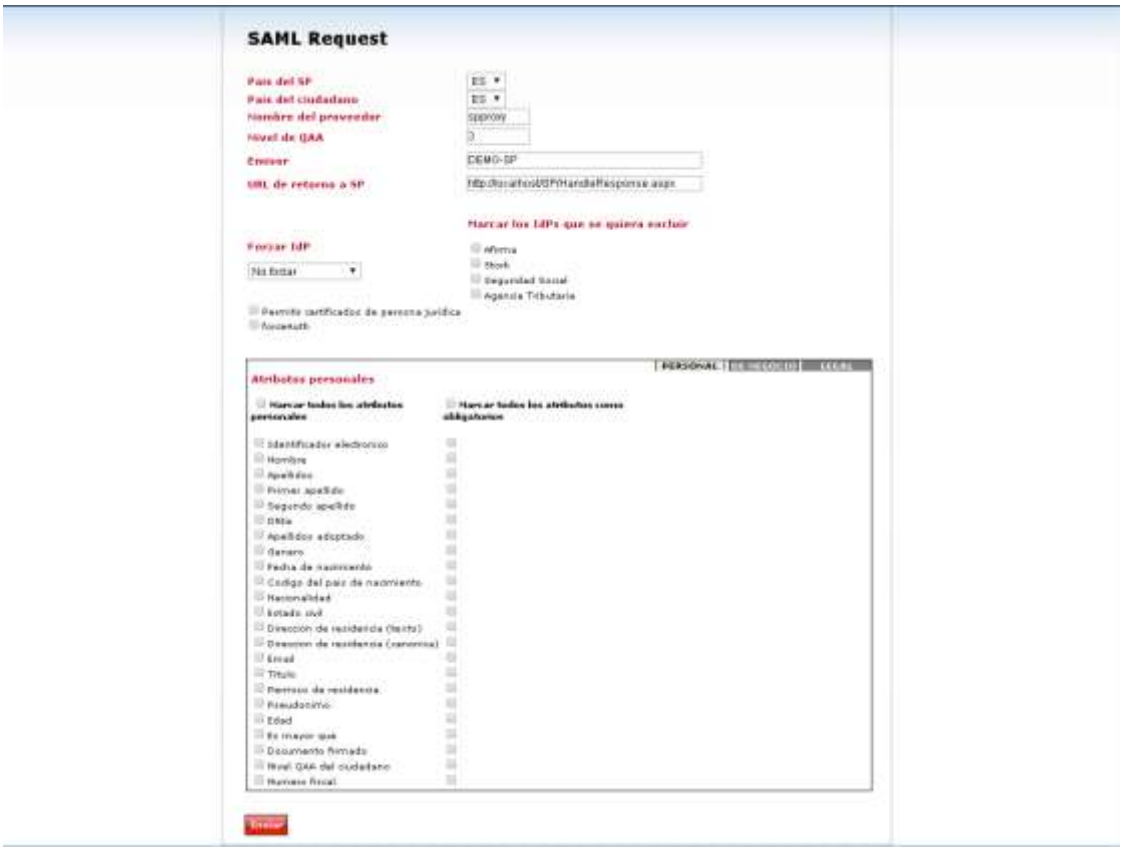


Figura 2: Página para personalizar petición - DemoSP:NET

En esta ventana se pueden seleccionar los atributos que se quieran solicitar a la pasarela (actualmente están disponibles los atributos de tipo "personal"), así como los IdPs que se van a utilizar. Para esto último existe una lista desplegable con los IdPs configurados para el caso en el que se quiera forzar a la pasarela a que use un IdP en concreto. En caso de que el IdP no se quiera forzar está disponible otra opción de filtrado en forma de checkboxes. Los IdPs que sean marcados serán excluidos en la consulta, de forma que el usuario no podrá seleccionarlos como Proveedores de Identidad.

Cuando un ciudadano accede a la plataforma Cl@ve, el proveedor de servicio hará una petición de datos del usuario, sobretodo cuando se trata de la primera vez que accede. Estos datos se extraen de sus credenciales o de bases de datos verificadas y mantenidas por las autoridades competentes, de tal manera que el proveedor de servicios pueda fiarse totalmente de los datos recibidos. Además, la calidad de estos datos está ligada al nivel de garantía de calidad de las credenciales requeridas por el proveedor de servicios; algunos de ellos pueden solicitar unos datos de calidad alta, mientras que otros se conformarán con un nivel medio o más bajo.

El proveedor de servicios se basa en los resultados obtenidos de la autenticación online para establecer la identidad de un subscriptor/usuario para realizar la transacción. El proveedor de servicios y el verificador pueden ser la misma entidad o pueden ser entidades diferentes. Si son entidades diferentes, el proveedor de servicios recibe una confirmación por parte del verificador.

El Proveedor de Servicios es responsable de validar que la confirmación vino de un verificador de confianza. Cuando estas confirmaciones indican la fecha de su creación o atributos asociados al demandante, el Proveedor de Servicios es también responsable de verificar esta información.

El Proveedor de Servicios determina que credenciales son requeridas para proporcionar al demandante o subscriptor acceso. Es por lo tanto el Proveedor de Servicios el que determina el nivel de autenticación para acceder a los datos.

Si la autenticación termina en éxito, deberá ver una tabla con los atributos pedidos junto con el valor obtenido (así como el nivel QAAy el IdP utilizado en la autenticación) tal y como se ve en la Figura 3.



Figura 3: Página de Retorno - DemoSP .NET

6 Single Sign On

La pasarela cuenta con un sistema de Single Sign On para que el usuario no tenga que re-autenticarse continuamente si va a realizar varias peticiones en un breve lapso de tiempo. Actualmente, si el tiempo entre peticiones es inferior a una hora el sistema reutilizará los datos que tiene almacenados en sesión relativos a ese usuario en lugar de obligarle a repetir todo el proceso de identificación.

En todo caso el sistema Cl@ve solicitará una nueva autenticación cuando se requiera un nivel QAA superior al nivel para el que el usuario ha sido autenticado previamente o cuando el SP fuerce un Proveedor de Identificación diferente.

6.1 Integración de la desconexión

El proceso de SSO es automático por parte de la pasarela, el SP no tiene que hacer nada para disponer de SSO. Sin embargo, el SP debe integrar el botón de desconexión o logout de su página con la desconexión del sistema cl@ve para cerrar la sesión en ambos entornos. Cerrar la sesión solamente en la página del SP realmente no desconecta al usuario ya que la conexión es automática a través de la pasarela Cl@ve y en ésta la sesión seguiría abierta. En el SP Demo se ofrece un ejemplo de cómo integrar la desconexión de la página con la desconexión en Cl@ve.

6.2 Forzar la autenticación

En caso de que el Proveedor de Servicios quiera obligar al usuario a que se identifique siguiendo el proceso completo al margen de que ya tuviese una sesión previa (y vigente) en la pasarela, el SP debe mandar el parámetro “ForceAuthn” con el valor “true” en la SAMLRequest que se le envía a la pasarela. De esta forma la pasarela ignorará la sesión previa del usuario y le forzará a realizar el proceso de autenticación como si fuese la primera vez que accede a la plataforma.

Forzar la autenticación tiene el efecto de reiniciar la sesión SSO. Ello significa que para los SP que no fuercen la autenticación la sesión SSO seguirá disponible.

En la Figura 2 se puede apreciar el checkbox “forceAuth” encima de los atributos a seleccionar para enviar. A la hora de preparar la petición SAML se introduce el parámetro “forceAuthn” con el valor “true”.

7 SAML Engine API

7.1 Ejemplos de uso

7.1.1 Generando una petición de autenticación automática

En este caso se genera la petición al crear la página. Ésta es asociada a un formulario que se envía al pulsar el botón "Mandar petición". En caso de que se quiera ver la petición que va a ser enviada puede hacerse a través del código html de la página.

7.1.2 Generando una petición de autenticación a medida

```
//Creating STORK Authentication Request --------------------------------
SAMLRequest request = new SAMLRequest();

//Filling Authentication Request fields --------------------------------
request.Destination = (…);
request.AssertionConsumerServiceURL = (…);
request.Country = (…);
request.ProviderName = (…);
request.Issuer = (…);
request.QAALevel = (…);
request.Id = (…);

//Loading Stork attributes to request
CitizenAttributes citizenAttributes = CitizenAttributes.Instance;
foreach (string personalName in citizenAttributes.GetPersonalNames()) {
    try {
        if (cbListPersonalAttributes.Items.FindByValue(personalName).Selected)
            request.AddAttribute(personalName,

cbListPersonalRequired.Items.FindByValue(personalName).Selected);
    } catch (Exception) {
    }
}

//Getting STORKSAMLEngine object  ---------------------------------------
SAMLEngine samlEngine = SAMLEngine.Instance;
samlEngine.Init(HttpContext.Current.Server.MapPath(".") + "/bin/");
XmlDocument xml = samlEngine.GenerateRequest(request);
Convert.ToBase64String(Encoding.UTF8.GetBytes(xml.OuterXml));
```

Este *SAMLRequest* debe ser colocado dentro de un formulario HTML. Por ejemplo, puede usar el método *WebUtils' PreparePOSTForm* en el método *Custom.aspx.cs' do_POST*:

```
postForm = WebUtils.PreparePOSTForm(
                        samlRequestField, relayStateField, countryField,
                        TextBoxSAMLrequest.Text, "State information to be
persisted across", countryList.SelectedValue,
DropDownListCPEPSCountry.SelectedValue, idpList);
```

7.1.3 Validando y leyendo una respuesta de autenticación

```
byte[] reqDataB64 = Convert.FromBase64String(requestData);
    string reqData = Encoding.UTF8.GetString(reqDataB64);
```

```
    XmlDocument xml = new XmlDocument();
    xml.PreserveWhitespace = true;
    xml.LoadXml(reqData);

    SAMLEngine.Instance.Init(HttpContext.Current.Server.MapPath(".") + "/bin/");
    SAMLResponse samlResponse = SAMLEngine.Instance.HandleResponse(xml);
```

```
public SAMLResponse HandleResponse(XmlDocument xmlResponse)
{
    try
    {
        int errorCode;
        SAMLResponse response;
        if ((errorCode = Verify(xmlResponse, null)) < 0)
            response = new SAMLResponse(errorCode);
        else
            response = ExtractResponseValues(xmlResponse);

        return response;
    }
    catch (Exception ex)
    {
        throw new SAMLException("EXCEPTION HandleResponse", ex);
    }
}
```

8 ANEXO: Petición de Alta como Proveedor de Servicios con acceso a la Plataforma Cl@ve

Fecha_Petición	*Fecha actual*
Nombre	*Nombre del Proveedor de Servicio*
Descripción	
Entorno	*PRE o PRO*
Persona_Contacto	*Persona de Contacto del Proveedor de Servicio (teléfono y email)*
Cerificado_SP	*Parte pública del certificado usado por el Proveedor de Servicio en la firma SAML*
-----BEGIN CERTIFICATE----- -----END CERTIFICATE-----	
Fecha_ prevista	*Fecha de Inicio prevista para pruebas en el entorno indicado*

Nota que nombre del proveedor de servicio y el certificado de firma tienen que coincidir con los valores configurados en 4.1.1.

mailto:stork@indra.es