Web Services SeguriSign Tools

versión 2.8

Manual de Usuario

Revisión 1.0

SeguriData

SeguriData Privada, S.A. de C.V.

Av. Insurgentes Sur #2375, 3er. piso,
Col. Tizapán, Del. Alvaro Obregón,
C.P. 01000, México, D.F.

Tel. +52 (55) 3098-0700
Fax. +52 (55) 3098-0702

http://www.seguridata.com

Contenido

CAPÍTULO 1

¿Cómo utilizar este manual?

1.1 Simbología y convenciones

En todo el manual se hace uso de una simbología específica para hacer más sencilla la identificación del tipo de información que se expone, así como de convenciones tipográficas, para hacer más clara la documentación.

1.1.1 Recomendaciones y Advertencias

En los lugares que resulte mas oportuno, se insertarán comentarios sobre el contenido del texto.

Importante

Este tipo de anotaciones contienen sugerencias y aclaraciones que facilitan el uso de la aplicación.

Precaución

Este tipo de anotaciones advierten sobre posibles riesgos en las operaciones descritas en el texto y que pueden causar pérdida de funcionalidad o datos.

1.2 Objetivo del Manual

Explicar los procedimientos relacionados con la instalación y configuración de SeguriSign Tools.

1.3 Objetivo del Producto

Proporcionar Web Services que trabajan de forma conjunta con SeguriSign v7.0, para firma unilateral de documentos (con y sin contenido), verificar firmas unilaterales de documentos (con y sin contenido), uso de múltiples llaves de firma y generación de evidencias imprimibles para documentos con firmas unilateral y multilateral.

CAPÍTULO 2

Instalación de SeguriSign Tools

El siguiente capítulo explica cómo se debe llevar acabo la instalación de SeguriSign Tools versión 2.8. Las actividades enumeradas deberán ser realizadas por personal que cuente con los conocimientos sobre instalación y administración de sistemas, además de tener una cuenta con privilegios de administrador en el sistema.

Se asume que la persona que realizará la instalación es el administrador que cuenta con privilegios de acceso equivalentes y cuenta con conocimientos acerca de:

- el uso de programas bajo Windows, y
- instalación y configuración de bases de datos

2.1 Requerimientos de Hardware

Los requerimientos para el equipo en que se hará la instalación, son:

- Procesador Pentium IV @3GHz
- Espacio en disco duro 50MB
- 1GB en RAM
- Unidad lectora de CD-ROM
- Conexión a una red

2.2 Requerimientos de Software

Los requerimientos de software para el equipo en que se hará la instalación, son:

- Sistema Operativo Microsoft® Windows 2000, Windows 2003 o Windows XP; base de datos SQL Server 2000 (Para procesar archivos de Office 2003, se requiere tener instalado Office 2003)

- Tener instalada la aplicación EasyPDF y configurada como se indica en su manual de usuario
- Tener instalado cualquier lenguaje de programación que tenga soporte para consumir Web Services
- Tener instalado Adobe Acrobat Reader

2.3 Ejecución del Instalador de SeguriSign Tools

Para iniciar la instalación deberá contar con el CD de la aplicación. Se recomienda leer el procedimiento de instalación para asegurarse de que se cuenta con todos los elementos necesarios para la instalación, así como tenerlo a la mano como referencia.

Finalmente, para el eventual caso de que requiera apoyo del Soporte Técnico de SeguriData, encontrará nuestros teléfonos en las primeras páginas de este manual.

Inserte el CD de la aplicación en el lector de CD-ROM y con ayuda del explorador de archivos localice el archivo SignToolsWS.exe. Ejecútelo haciendo doble clic en el nombre del archivo.

A continuación se iniciará el asistente de instalación de SeguriSign Tools. (Figura 2.1).

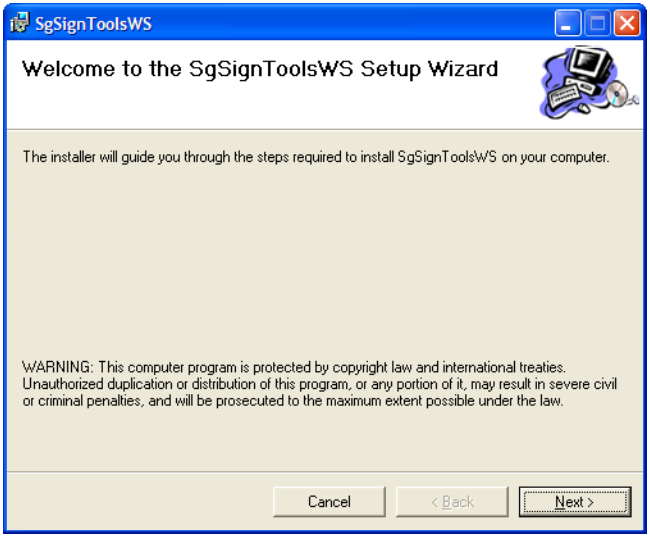

Figura 2.1 Inicio del Setup Wizard de SeguriSign Tools

Para comenzar la instalación, haga clic en el botón Next >.

Indique la ubicación del folder donde se instalará SeguriSign Tools, como se muestra en la Figura 2.2.

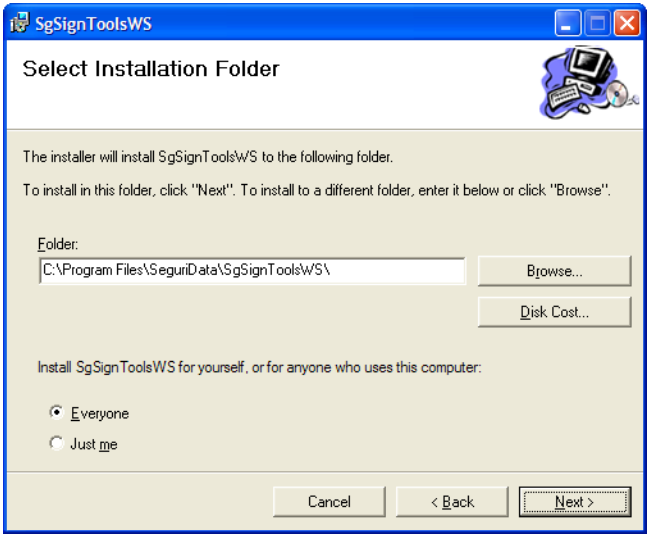

Figura 2.2 Selección de la Carpeta de Instalación

El valor predeterminado es:

C:\Program Files\SeguriData\SgSignToolsWS

sin embargo puede cambiarse libremente la ubicación por otra ruta válida, sin que esto repercuta en el desempeño de la aplicación.

Seleccione la opción Everyone si desea que todos los usuarios puedan utilizar SeguriSign Tools en el equipo donde se instaló la aplicación, o seleccione Just me si desea que solo usted pueda utilizar SeguriSign Tools.

Para continuar con el proceso, deberá de hacer clic en el botón Next >.

En la siguiente ventana se presenta la confirmación de la instalación. (Figura 2.3).

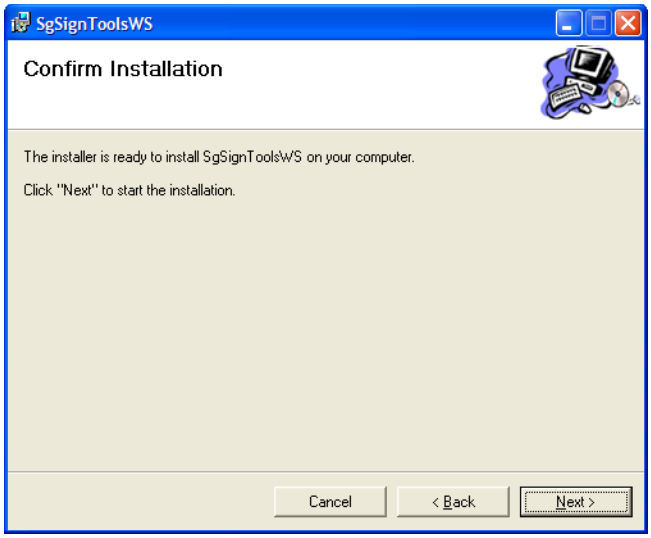

Figura 2.3 Confirmación de la Instalación

Para continuar con la instalación, haga clic en el botón Next >.

El proceso de copia de archivos se iniciará mostrando el avance de la instalación. (Figura 2.4).

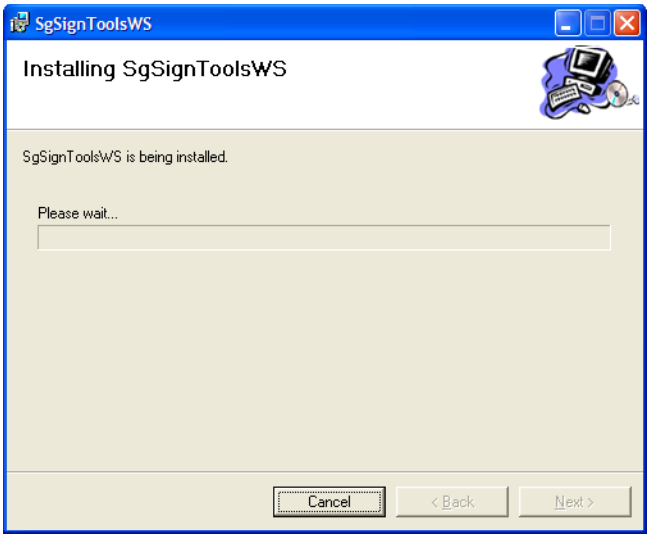

Figura 2.4 Avance de la Instalación

Espere unos segundos.

A continuación se presentará el resultado de la instalación de SeguriSign Tools. (Figura 2.5).

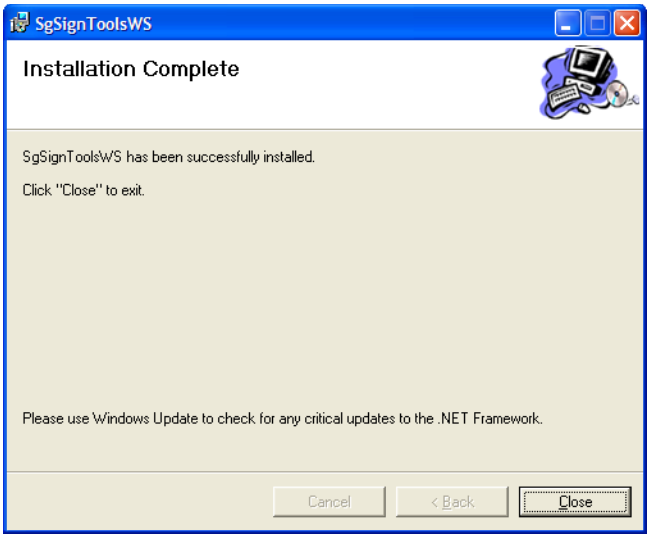

Figura 2.5 Resultado de la Instalación

Por último, haga clic en el botón Close.

2.4 Creación de la Base de Datos

Para la creación de la base de datos siga los siguientes pasos:

1. Crear la base de datos con el script que se encuentra en la ruta de instalación: ...\BD\SgSignTools.sql
2. Crear un usuario para la base de datos.

2.5 Instalación del Servicio

1. Entrar a la ruta de instalación:
   C:\Program Files\SeguriData\SgSignToolsWS\
   desde línea de comandos y ejecutar la siguiente línea:
   SgSignToolsWS.exe /registerservice
2. Establecer los parámetros en el archivo de configuración. El archivo de configuración se encuentra en la ruta de instalación y se llama SgSignToolsWS.properties
3. Las propiedades que debe establecer son las siguientes:
   3.a El puerto donde estará trabajando el servicio SgSignToolsWS
   Server.1.port=8081

3.b La dirección IP donde se encuentra SeguriSign 7:
SeguriSign.ip=192.168.0.185

3.c El puerto de SeguriSign 7: SeguriSign.port=7979

3.d Una lista con las extensiones de los archivos para lo que se pueden generar evidencia imprimible: ConvertFileExt=doc|xls|ppt|txt|rtf

3.e La ruta del archivo de marca de agua:
WaterMark_Config=${application.dir}WaterMark/
PDF_WaterMark_Config.config

3.f La cadena de conexión a la base de datos:
DB.connectionString=USER=UserTools,PASSWORD=12121212qw, DBNAME=
192.168.0.35@SgSignTools

3.g Un indicador del manejador de base de datos: 0 para SQL SERVER y 1 para Oracle. DB.TYPE=0

4. Entrar a la ruta de instalación:
C:\Program Files\SeguriData\SgSignToolsWS\
desde línea de comandos y ejecutar la siguiente línea:

```
SgsignToolsWS.exe /Initialize /Administrator="path_certificado_del_
administrador.cer" /ServiceP12="llaves_del_servicio.p12" /
ServicePWD="password_del_p12"
```

donde:

- Path_certificado_del_administrador.cer: ruta del certificado del administrador de los Web services.
- Llaves_del_servicio: certificado .p12 o .pfx que trae las llaves pública y privada que corresponden al servicio.
- Password_del_p12: password del certificado del Servicio.

5. Asignación de Water Mark por default
   - Water Mark Default = 1
   - En caso de no querer aplicar una Marca de Agua por default comentar esta línea.
6. Iniciar el servicio.

CAPÍTULO 3

Web Services de SeguriSign Tools

A continuación se detallan las operaciones disponibles en el Web Service de SeguriSign.

Agregar usuarios que accedan al Web service

Esta operación permite agregar usuarios que posteriormente podrán utilizar el Web Service. Usuario administrador y usuario normal.

El usuario *administrador* podrá agregar otros usuarios, agregar llaves y asignar llave a usuario, además de consumir las operaciones de firma, verificación y generación de evidencias. El usuario que *no es administrador* sólo podrá consumir las operaciones de firma, verificación y generación de evidencias.

Entrada:

- El parámetro email ponerlo como opcional.
  - ANTES: email (tipo string**.** El correo electrónico del usuario)
  - NUEVO: email **[opcional]** (tipo string**.** El correo electrónico del usuario)

Propiedades que debe establecer en *AddUserRequest:*

- certificate [**opcional**] (tipo SgSignWS.Document(). Certificado con que el usuario consumirá el Web service usando SSL)
  - compressed (tipo bool. Indica si los datos van comprimidos)
  - data (tipo byte [**arreglo**]. Es el archivo en forma de arreglo de bytes)
- email (tipo string. El correo electrónico del usuario**)**

- isActive (tipo bool**.** Indica si el usuario se agregará como usuario activo). Si no se establece como activo, quedará como falso.
- isAdministrador (tipo bool. Indica si el usuario es administrador). Si no establece si es administrador, quedará como falso.
- password (tipo string. Password del usuario)
- username (tipo string. Nombre del usuario)

Salida

Propiedades que regresa *AddUserResponse:*

- status (tipo string. Ok en caso de que se agregue el usuario correctamente)

Importante

En caso de que suceda un error, el Web service lanzará una excepción con la información existente del error.

Ejemplo:

```
[1]  try
[2]  {
[3]  SgSignWS.SgSignToolsWS ws = new SgSignWS.SgSignToolsWS();
[4]  ws.Url = "http://196.168.0.35:8081";
[5]  ws.Credentials = new NetworkCredential("admin", "23232323we");
[6]  SgSignWS.AddUserRequest req = new SgSignWS.AddUserRequest();
[7]  byte[] bytesCertificate = null;
[8]  try
[9]  {
[10]     bytesCertificate = File.ReadAllBytes(txtAddUser_Cer.Text);
[11] }
[12] catch (Exception exep)
[13] {}
[14] if (bytesCertificate != null)
[15] {
[16]    req.certificate = new SgSignWS.Document();
[17]    req.certificate.compressed = false;
[18]    req.certificate.data = bytesCertificate;
[19] }
[20] req.email = txtAddUser_Mail.Text;
[21] req.isActive = chkAddUser_IsActive.Checked;
[22] req.isAdministrador = chkAddUser_IsAdmin.Checked;
[23] req.password = txtAddUser_Pwd.Text;
[24] req.username = txtAddUser_UserName.Text;
[25] SgSignWS.AddUserResponse resp = SgSignWS.AddUserResponse)ws.ProcessMessage(req);
[26] MessageBox.Show("Se agregó un USUARIO = " +resp.status );
[27] }
```

```
[28]  catch (Exception exep)
[29]  {
[30]     MessageBox.Show(exep.Message);
[31]  }
```

La Figura 3.1 muestra un ejemplo de excepción al intentar llamar la operación *AddUserRequest* en donde el nombre de usuario no se estableció.

Figura 3.1 Excepción al Agregar Usuario

Cuando el usuario consuma los Web Services deberá capturar las excepciones y manejarlas. Un ejemplo de una excepción manejada se muestra en la Figura 3.2. Es un ejemplo de excepción al intentar llamar la operación *AddUserRequest* en donde el nombre de usuario no se estableció.

Figura 3.2 Excepción Manejada al Agregar Usuario

Agregar llaves que permitan firmar

Esta operación permite agregar llaves que se usan para firmar documentos.

Puede ser de dos formas: ya sea usando un certificado que contenga las dos llaves (.p12 o .pfx), o agregando cada llave por separado (.cer y .key)

Entrada:

- El parámetro description ponerlo como opcional
  - ANTES: description (tipo string. Descripción de la llave)
  - NUEVO: description [**opcional**] (tipo string. Descripción de la llave)

Propiedades que debe establecer en *AddKeyRequest*:

- certificate (tipo byte [**arreglo**]). Es el certificado para firmar documentos en forma de arreglo de bytes, si es un .cer se requiere establecer la key, y si es un .p12 o .pfx la propiedad key ya no se debe establecer.
- cryptoengine (tipo SgSignWS.CryptoEngine). Indica si se usará el motor criptográfico de seguridata o el de windows. Si el usuario no establece ninguna, el valor default es SGLIB.
- description (tipo string). Descripción de la llave.
- key [**opcional**] (tipo byte [**arreglo**]). Es la llave para firmar documentos en forma de arreglo de bytes, se requiere establecerla sólo si el certificado es .cer.
- password (tipo string). Password del usuario.

Salida

Propiedades que regresa *AddKeyResponse*:

- keyID (tipo string**.** Identificador que quedó asociado con la llave que se acaba de agregar)

Importante

En caso de que suceda un error, el Web Service lanzará una excepción con la información existente del error.

Ejemplo:

```
[1] SgSignWS.SgSignToolsWS ws = new SgSignWS.SgSignToolsWS();
[2] ws.Url = "http://196.168.0.35:8081";
[3] ws.Credentials = new NetworkCredential("admin", "12121212");
[4]
[5] SgSignWS.AddKeyRequest req = new SgSignWS.AddKeyRequest();
[6] req.certificate = File.ReadAllBytes(txtCerKey.Text);
[7] if(rbnCrypto.Checked == true)
[8] req.cryptoengine =  SgSignWS.CryptoEngine.CAPI;
```

```
[9]  else if(rbnSglib.Checked == true)
[10] req.cryptoengine = SgSignWS.CryptoEngine.SGLIB;
[11] req.description = txtDescKey.Text;
[12] req.key = File.ReadAllBytes(txtKeyAdd.Text);
[13] req.password = txtPwdKey.Text;
[14]
[15] SgSignWS.AddKeyResponse resp = (SgSignWS.AddKeyResponse)ws.ProcessMessage(req);
[16] string s = resp.keyID;
[17] MessageBox.Show("Se agregó una llave con Id = " + s);
```

Asignar llave

Esta operación permite asignar a un usuario una llave con la que podrá firmar documentos.

Cada usuario podrá usar varias llaves. Cada llave podrá ser usada por varios usuarios

Entrada:

Propiedades que debe establecer en *AssignKeyToUserRequest*:

- keyID (tipo string). Identificador de la llave.
- username (tipo string). Nombre del usuario al que se asignará la llave.

Salida

Propiedades que regresa *AssignKeyToUserResponse*:

status (tipo string). Ok en caso de que se asigne la llave correctamente**.**

Importante

En caso de que suceda un error, el Web service lanzará una excepción con la información existente del error.

Ejemplo:

```
[1]  SgSignWS.SgSignToolsWS ws = new SgSignWS.SgSignToolsWS();
[2]  ws.Url = "http://196.168.0.35:8081";
[3]  ws.Credentials = new NetworkCredential("admin", "12121212");
[4]
[5]  SgSignWS.AssignKeyToUserRequest req = new SgSignWS.AssignKeyToUserRequest();
[6]  req.keyID = txtAsignIdKey.Text;
[7]
[8]  req.username = txtAsignUser.Text;
[9]
[10] SgSignWS.AssignKeyToUserResponse resp =
[11] (SgSignWS.AssignKeyToUserResponse)ws.ProcessMessage(req);
[12] string s = resp.status;
[13] MessageBox.Show("Se asignó una llave = " + s);
```

Firma unilateral de documentos

Esta operación permite firmar un documento.

Entrada

Propiedades que debe establecer en *SignDocumentRequest*:

- docToSign (tipo SgSignWS.Document()). Documento que se va a firmar.
  - compressed (tipo bool). Indica si los datos van comprimidos.
  - data (tipo byte [**arreglo**]). Es el archivo en forma de arreglo de bytes.
  - filename (tipo string)**.** Nombre del documento que se va a firmar**.**
- keyID (tipo string). Identificador de la llave con que se va a firmar.
- withContent (tipo bool). Indica si el pkcs7 incluye contenido.

Salida

Propiedades que regresa *SignDocumentResponse*:

- signedDoc (tipo SgSignWS.Document()). Documento firmado.
  - compressed (tipo bool). Indica si los datos van comprimidos.
  - data (tipo byte [**arreglo**]) . Es el archivo en forma de arreglo de bytes.
  - filename (tipo string). Nombre del documento que se va a firmar.
- status (tipo string). Ok en caso de que se asigne la llave correctamente**.**

Importante

En caso de que suceda un error, el Web service lanzará una excepción con la información existente del error.

Ejemplo:

```
[1]  SgSignWS.SgSignToolsWS ws = new SgSignWS.SgSignToolsWS();
[2]  ws.Url = "http://196.168.0.35:8081";
[3]  ws.Credentials = new NetworkCredential(txtSignUser.Text, txtSignPwd.Text);
[4]  SgSignWS.SignDocumentRequest req = new SgSignWS.SignDocumentRequest();
[5]
[6]  req.docToSign = new SgSignWS.Document();
[7]  req.docToSign.filename = Path.GetFileName(txtFindToSign.Text);
[8]  req.docToSign.compressed = false;
[9]  req.docToSign.data = File.ReadAllBytes(txtFindToSign.Text);
[10] req.keyID = txtKeyToSign.Text;
[11] req.withContent = true;
[12]
[13] SgSignWS.SignDocumentResponse resp =
[14] SgSignWS.SignDocumentResponse)ws.ProcessMessage(req);
```

```
[15]  bool status = resp.status;
[16]  File.WriteAllBytes(txtSetSignedFile.Text, resp.signedDoc.data);
[17]  MessageBox.Show("Se firmó un archivo CON contenido");
```

Verificación de Firma unilateral de documentos usando SeguriSign versión 7

Esta operación permite verificar la firma unilateral de documentos utilizando SeguriSign versión 7.

Entrada

Propiedades que debe establecer en *VerifyRequest*:

- originalDoc [**opcional**] (tipo SgSignWS.Document()). Documento original que se firmó. Este parámetro se requiere sólo cuando se verifica una firma que no incluye el contenido.
  - compressed (tipo bool). Indica si los datos van comprimidos.
  - data (tipo byte [**arreglo**]). Es el archivo en forma de arreglo de bytes.
  - filename (tipo string). Nombre del documento original que se firmó.
- signedDoc (tipo SgSignWS.Document()). Documento que se va a verificar.
  - compressed (tipo bool). Indica si los datos van comprimidos.
  - data (tipo byte [**arreglo**]). Es el archivo en forma de arreglo de bytes.
  - filename (tipo string). Nombre del documento original que se firmó.

Salida

Propiedades que regresa *VerifyResponse*:

- tspDoc (tipo SgSignWS.Document()). Documento que contiene la estampilla de tiempo del momento de la firma.
  - compressed (tipo bool). Indica si los datos van comprimidos.
  - data (tipo byte [**arreglo**]). Es el archivo en forma de arreglo de bytes.
  - filename (tipo string). Nombre del documento que se va a firmar.
- sequence (tipo string). Identificador con el que se asocia de manera única este documento al servidor de SeguriSign.

Importante

En caso de que suceda un error, el Web service lanzará una excepción con la información existente del error.

Ejemplo:

```
[1] SgSignWS.SgSignToolsWS ws = new SgSignWS.SgSignToolsWS();
[2] ws.Url = "http://196.168.0.35:8081";
[3] ws.Credentials = new NetworkCredential(txtVerifyWithoutUser.Text, txtVerifyWithoutPwd.Text);
[4]
[5] SgSignWS.VerifyRequest req = new SgSignWS.VerifyRequest();
[6] req.originalDoc = new SgSignWS.Document();
[7] req.originalDoc.filename = Path.GetFileName(txtFindOriginal.Text);
```

```
[8]  req.originalDoc.compressed = false;
[9]  req.originalDoc.data = File.ReadAllBytes(txtFindOriginal.Text);
[10]
[11] req.signedDoc = new SgSignWS.Document();
[12] req.signedDoc.filename = Path.GetFileName(txtFindSignedWithout.Text);
[13] req.signedDoc.compressed = false;
[14] req.signedDoc.data = File.ReadAllBytes(txtFindSignedWithout.Text);
[15]
[16] SgSignWS.VerifyResponse resp = (SgSignWS.VerifyResponse)ws.ProcessMessage(req);
[17] File.WriteAllBytes(txtSetTSPwithout.Text, resp.tspDoc.data);
[18] MessageBox.Show("Se verificó un archivo sin contenido\n" + "Secuencia=" +
[19] resp.sequence);
[20] txtSequenceWithout.Text = resp.sequence;
```

Generar la evidencia imprimible de archivos con firma unilateral

Esta operación permite generar una evidencia imprimible de documentos con firma unilateral.

Las evidencias se podrán generar para los archivos que tengan instalado el software que los interpreta y que además soporten el comando "PRINT" del sistema operativo (por ejemplo: para los archivos de Office, se deberá tener instalado Office). La evidencia se conforma con los siguientes elementos:

- El contenido del documento original se convierte a PDF.
- El identificador que tiene ese documento en SeguriSign.
- *Del firmante*: se agregan los siguientes datos: nombre, número de serie del certificado, un indicador si el certificado está vigente y un indicador si el certificado está revocado.
- *De la firma:* se agregan los siguientes datos: fecha y hora de la firma, el algoritmo de encripción, la cadena de firma. Se revisa si el certificado del firmante sea vigente, además se extrae el nombre del firmante y el número de serie del certificado.
- *Del respondedor OCSP:* se agregan los siguientes datos: fecha y hora de la consulta al respondedor OCSP, nombre del respondedor, nombre de la autoridad certificadora que emitió el respondedor y número de serie del certificado que se validó en la transacción.
- *Del respondedor TSP:* se agregan los siguientes datos: fecha y hora de la consulta al respondedor TSP, nombre del respondedor, nombre de la autoridad certificadora que emitió el respondedor, secuencia que se tiene del documento en SeguriSign y datos estampillados.

Entrada

Propiedades que debe establecer en *GetPrintableUnilateralRequest*:

- watermark_id (tipo Integer)
- idFromVerify (tipo string. Identificador que se obtuvo al verificar este documento en SeguriSign)

Salida

Propiedades que regresa *GetPrintableUnilateralResponse*:

- printableDoc (tipo SgSignWS.Document()). Documento que contiene el documento pdf que se genera como evidencia. Con una sola firma.
  - compressed (tipo bool). Indica si los datos van comprimidos.
  - data (tipo byte [**arreglo**]). Es el archivo en forma de arreglo de bytes.
  - filename (tipo string). Nombre del documento de evidencia.

> **Importante**
>
> En caso de que suceda un error, el Web service lanzará una excepción con la información existente del error

Ejemplo:

```
[1]  SgSignWS.SgSignToolsWS ws = new SgSignWS.SgSignToolsWS();
[2]  ws.Url = "http:// 196.168.0.35:8081";
[3]  ws.Credentials = new NetworkCredential(txtPrintableUser.Text, txtPrintablePwd.Text);
[4]
[5]  SgSignWS.GetPrintableUnilateralRequest req = new SgSignWS.GetPrintableUnilateralRequest();
[6]
[7]  req.idFromVerify= txtSequenceToPrint.Text;
[8]  SgSignWS.GetPrintableUnilateralResponse resp =
[9]  SgSignWS.GetPrintableUnilateralResponse)ws.ProcessMessage(req);
[10] File.WriteAllBytes(@"D:\super.pdf", resp.printalbeDoc.data);
[11] MessageBox.Show("Se generaron las evidencias");
[12]
```

Se cuenta también con Get Printable Multilateral Request

> **Importante**
>
> En caso de que **no** se desee Marca de Agua usar el valor -1

A continuación se presentan los web services para la generación de archivos PDF's con distintas marcas de agua dependiendo del proceso que solicite la autenticación de la firma electrónica.

- class WaterMarkInfo
  - int watermark_id;
  - string name
- GetWaterMarksListRequest - Obtiene la lista de Marcas de Agua que tenga el servidor
- GetWaterMarksListResponse - Respuesta
  - arreglo de WaterMarkInfo - Arreglo
- AddWaterMarkRequest - Agrego una nueva Marca de Agua
  - name : Nombre asignado
  - watermarkDescFile : archivo con la configuracion
- AddWaterMarkResponse
  - watermark_id : respuesta ID asignado
- UpdateWaterMarkRequest

- watermark_id : ID a modificar
- name : nuevo Nombre
- watermarkDescFile : nuevo archivo de configuracion

- GetWaterMarkRequest
  - watermark_id : id de la marca de agua a obtener
- GetWaterMarkResponse
  - watermark_id : identificador
  - name : nombre
  - watermarkDescFile : archivo de configuracion
- DeleteWaterMarkRequest
  - watermark_id : ID de la marca de agua a borrar

3.1 Generación de usuarios para la Marca de Agua

En el directorio de instalación localice el archivo **portalserver.properties** y configurar la URL de los Web Services de SeguriSign

```
#
# OSP Configuration
#
osp.codeCache        = ${application.dir}codeCache
osp.bundleRepository = ${application.configDir}bundles
osp.data             = ${application.configDir}data


#
# Logging Configuration
#
logging.loggers.root.channel=  c1
logging.loggers.root.level= debug
logging.channels.c1.class= FileChannel
logging.channels.c1.path= ${application.dir}PortalServer.txt
logging.channels.c1.formatter.class= PatternFormatter
logging.channels.c1.formatter.pattern= %Y-%m-%dT%H:%M:%S : %I - [%p] %t
logging.channels.c1.formatter.times=local
```

```
# osp.web.server.port: The TCP port where the (non-secure) server is
listening. Defaults to 22080.

# osp.web.server.securePort: The TCP port where the secure (HTTPS) server
is listening. Defaults to 22443.

# osp.web.server.maxQueued: The maximum number of queued requests (see
Poco::Net::TCPServerParams). Defaults to 100.

# osp.web.server.maxThreads: The maximum number of threads used by the
server (see Poco::Net::TCPServerParams


osp.web.server.port=23080

osp.web.server.securePort=23443

#

# NetSSL (OpenSSL) Configuration

openSSL.server.privateKeyFile = ${application.configDir}any.pem

openSSL.server.caConfig = ${application.configDir}rootcert.pem

openSSL.server.verificationMode = none

openSSL.server.verificationDepth = 9

openSSL.server.loadDefaultCAFile = false

openSSL.server.cypherList = ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH

openSSL.server.privateKeyPassphraseHandler.name = KeyFileHandler

openSSL.server.privateKeyPassphraseHandler.options.password = test

openSSL.server.invalidCertificateHandler = AcceptCertificateHandler
```

WS.url=http://192.168.0.229:8181

HTTP://IP:PUERTO

Guarde los cambios de la configuración

Abra un navegador y coloque la URL y Puerto del Portal de Administración. (Figura 3.3).

Figura 3.3 Inicio del Portal de Administración SeguriSign Tools WS

Proporcione el usuario y password con los cuales de inicializo la base de datos.

En seguida ventana debe de elegir un estado y la opción de administradores. (Figura 3.4).

Figura 3.4 Portal de Administración SeguriSign Tools WS

(Figura 3.5).

Figura 3.5 Portal de Administración SeguriSign Tools WS/ Administradores

3.2 Generación de archivos PDF's con distintas marcas de agua

3.2.1 Primer paso

Definir los perfiles asociados con las distintas marcas de agua

3.2.2 Segundo Paso

A continuación se debe de ejecutar la aplicación de Water Mark Config (Figura 3.6).

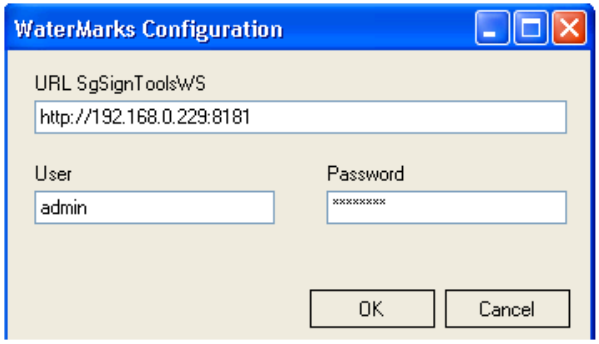

Figura 3.6 WaterMarks Configuration

3.2.3 Tercer Paso

Proporcione la URL de los Web Services de SeguriSign, usuario y password.

Importante

El usuario debe ser un administrador para realizar dichos cambios.

3.2.4 Cuarto Paso

Con el botón **Nuevo** se solicita el nombre del nuevo perfil, solicitando la imagen en caso de tenerla y el texto de la marca de agua (Figura 3.7).

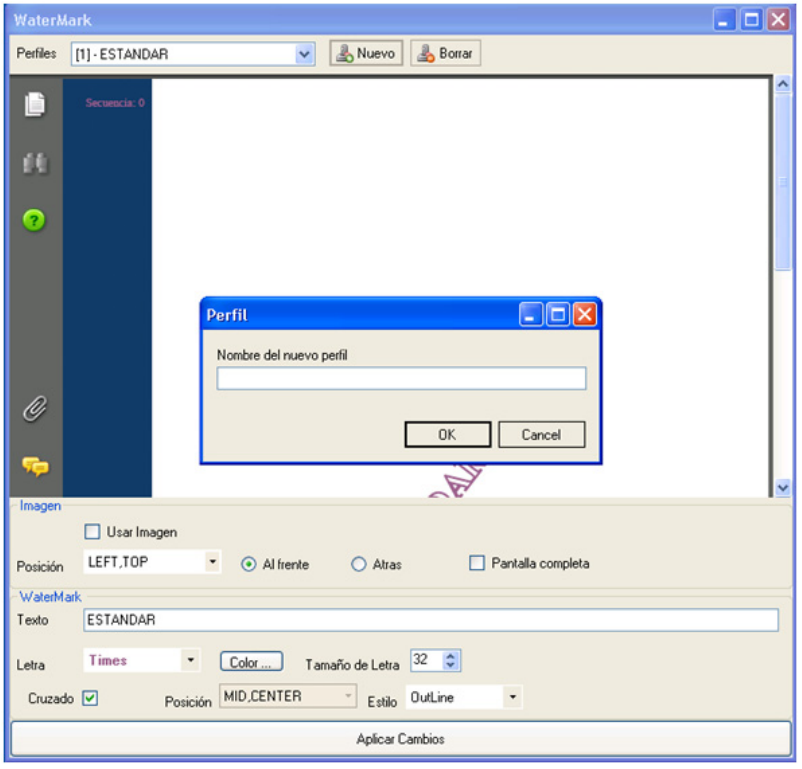

Figura 3.7 Botón Nuevo

Combo-Box Perfiles :aparecerá la lista de Marcas de Agua almacenadas en el servidor.

- Botón Nuevo: Agregar una Nueva Marca de Agua
- Botón Borrar : Eliminar la marca de agua que se encuentre seleccionada
- Check-Box Usar Imagen : al Seleccionarlo pedirá un archivo de Imagen para incorporar a la
- marca de Agua
- Botón Aplicar Cambios : Aplica los cambios realizador y los guarda en el servidor

(Figura 3.8).

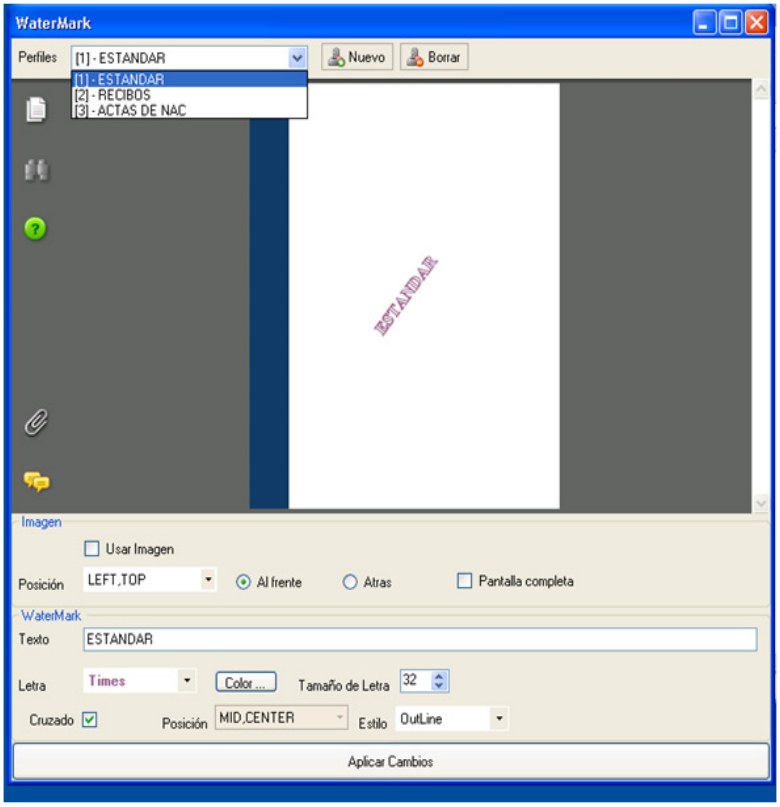

Figura 3.8 Marca de Agua

Importante

Las marcas de agua se generarán usando la aplicación WaterMarkConfig.exe, que se encuentra en la ruta de instalación.

Generar la evidencia imprimible de archivos con firma multilateral

Esta operación permite generar una evidencia imprimible de documentos con firma multilateral.

Entrada

Propiedades que debe establecer en *GetPrintableMultilateralRequest*:

- originalDoc (tipo SgSignWS.Document()). Documento original que se firmó.
  - compressed (tipo bool). Indica si los datos van comprimidos.
  - data (tipo byte [**arreglo**]). Es el archivo en forma de arreglo de bytes.
  - filename (tipo string). Nombre del documento original que se firmó.
- idFromInit (tipo string). Identificador que se obtuvo al inicializar este documento en SeguriSign.

Salida

Propiedades que regresa *GetPrintableMultilateralResponse*:

- printableDoc (tipo SgSignWS.Document()). Documento que contiene el documento pdf que se genera como evidencia. Con varias firmas.
  - compressed (tipo bool). Indica si los datos van comprimidos.
  - data (tipo byte [**arreglo**]). Es el archivo en forma de arreglo de bytes.
  - filename (tipo string). Nombre del documento de evidencia.

Importante

En caso de que suceda un error, el Web service lanzará una excepción con la información existente del error

Ejemplo:

```
[1]  SgSignWS.SgSignToolsWS ws = new SgSignWS.SgSignToolsWS();
[2]  ws.Url = " http:// 196.168.0.35:8081";
[3]  ws.Credentials = new NetworkCredential(txtPrintableUser.Text, txtPrintablePwd.Text);
[4]
[5]  SgSignWS.GetPrintableUnilateralRequest req = new SgSignWS.GetPrintableUnilateralRequest();
[6]  req.originalDoc = new PruebaWSdeC.SgSignWS.Document();
[7]  req.originalDoc.filename = Path.GetFileName(txtFindToPrint.Text);
[8]  req.originalDoc.compressed = false;
[9]  req.originalDoc.data = File.ReadAllBytes(txtFindToPrint.Text);
[10]
[11] req.idFromInit = txtSequenceToPrint.Text
[12]
[13] SgSignWS.GetPrintableUnilateralResponse resp = (SgSignWS.GetPrintableUnilateralResponse)ws.ProcessMessage(req);
[14] File.WriteAllBytes(@"D:\super.pdf", resp.printableDoc.data);
```